Call Center Capacity Planning

Nielsen, Thomas Bang

Publication date:
2010

Document Version
Publisher's PDF, also known as Version of record

Ph.D. Thesis

Call Center Capacity Planning

Thomas Bang Nielsen
DTU
Kgs. Lyngby
October 30, 2009

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD-2009-223

Preface

This thesis has been prepared at the Technical University of Denmark, Department of Informatics and Mathematical Modelling in partial fulfilment of the requirements for obtaining the Ph.D. degree in engineering. The thesis consists of a summary report and a collection of three research papers written and submitted for publication during the period 2007–2009. The Ph.D. was funded by a university grant from the Technical University of Denmark and carried out under the research school ITMAN.

First and foremost, I would like to thank my main supervisor Bo Friis Nielsen (DTU Informatics) for all his help during the three years and for believing in me. Also I would like to thank my co-supervisor Villy Bæk Iversen (DTU Fotonik) for his guidance. Additionally, I mention my colleagues at the Mathematical Statistics group, especially Jan Frydendall for sharing ups and downs in the office for the main part of the three years and Ellen-Marie Traberg-Borup for helping me with all the practical matters.

Next, I thank the operations management group at Danske Bank, in particular Mette Willemoes Berg for her cooperation and for helping me gain insight into the call center world.

I appreciate the OBP-group at the VU University Amsterdam for welcoming me in the fall of 2007. Especially I thank Ger Koole for letting me visit and for his invaluable help during the last years of my studies. Also thanks to Rene Bekker for his close cooperation on a paper and Sandjai Bhulai for letting me attend his class.

My family played an important role. In particular I would like to thank my parents Sonja and John for their support throughout my studies and my brothers Torben and Jesper for their interest in my work.

Last, but not least I thank my wonderful girlfriend Sara for always being there for me and making the last years much more enjoyable due to her love and cheerful mind.

October 2009
Thomas Bang Nielsen

Summary

The main topics of the thesis are theoretical and applied queueing theory within a call center setting.

Call centers have in recent years become the main means of communication between customers and companies, and between citizens and public institutions. The extensively computerized infrastructure in modern call centers allows for a high level of customization, but also induces complicated operational processes. The size of the industry together with the complex and labor intensive nature of large call centers motivates the research carried out to understand the underlying processes.

The customizable infrastructure allows customers to be divided into classes depending on their requests or their value to the call center operator. The agents working in call centers can in the same way be split into groups based on their skills. The discipline of matching calls from different customer classes to agent groups is known as skills-based routing. It involves designing the routing policies in a way that results in customers receiving a desired service level such as the waiting time they experience. The emphasis of this thesis is on the design of these policies.

The first paper, *Queues with waiting time dependent service*, introduces a novel approach to analyzing queueing systems. This involves using the waiting time of the first customer in line as the primary variable on which the analysis is based. The legacy approach has been to use the number of customers in queue. The new approach facilitates exact analysis of systems where service depends on the waiting time. Two such systems are analyzed, one where a server can adapt its service speed according to the waiting time of the first customer in line. The other deals with a two-server setup where one of the servers is only allowed to take customers who have waited a certain fixed amount of time. The latter case is based on a commonly used rule in call centers to control overflow between agent groups.

Realistic call center models require multi-server setups to be analyzed. For this reason, an approximation based on the waiting time of the first in line approach is developed in the paper *Waiting time dependent multi-server priority queues*, which is able to deal with multi-server setups. It is used to analyze a setup with two customer classes and two agent groups, with overflow between them controlled by a fixed threshold. Waiting time distributions are obtained in order to relate the results to the service levels used in call centers. Furthermore, the generic nature of the approximation is demonstrated by applying it to a system incorporating a dynamic priority scheme.

In the last paper *Optimization of overflow policies in call centers*, overflows between agent groups are further analyzed. The design of the overflow policies is optimized using Markov Decision Processes and a gain with regard to service levels is obtained. Also, the fixed threshold policy is investigated and found to be appropriate when one class is given high priority and when it is desired that calls are answered by the designated agent class and not by other groups through overflow.

Resumé

Hovedområderne for denne afhandling er teoretisk og praktisk køteori anvendt inden for call centre.

Call centre har i de senere år vundet frem som måden, hvorpå firmaer og offentlige institutioner kommunikerer med kunder og borgere. En omfattende computerstyret infrastruktur i moderne call centre giver mulighed for en høj tilpasningsevne, men kan i samme omgang gøre driften særdeles kompleks. Størrelsen og den arbejdstunge karakter af call center industrien, samt de komplekse underliggende processer, gør call centre til et interessant forskningsemne.

Den computerstyrede infrastruktur betyder, at kunder kan deles op i forskellige klasser alt efter grunden til deres henvendelse eller efter deres værdi for call center operatøren. På samme måde kan medarbejderne i call centre deles op i grupper efter deres kvalifikationer. Problemet med at parre forskellige kundeklasser og medarbejdergrupper omtales som *skills-based routing*. Dette omfatter dirigering af opkald til de rette medarbejdere under hensyntagen til den service kunderne skal have, herunder den ventetid de bliver udsat for. Denne afhandling omhandler design af disse regler for opkaldsdirigering.

Den første artikel, *Queues with waiting time dependent service*, introducerer en ny tilgang til analyse af køsystemer. Denne tilgang tager udgangspunkt i brugen af ventetiden for den forreste kunde i køen som den primære variabel analysen baseres på. Den traditionelle tilgang er at bruge antal kunder i kø som den primære variabel. Den nye tilgang gør det muligt at analysere systemer, hvor betjeningen afhænger af den tid, kunden forrest i køen har ventet. To sådanne systemer bliver analyseret. I det første kan et betjeningssted tilpasse betjeningshastigheden som funktion af den tid, den forreste kunde i køen har ventet. Det andet system, der analyseres, omfatter to servere, hvoraf den ene kun må tage kunder, der har ventet en given fast tid. Det sidste tilfælde er baseret på en regel, der ofte bruges i call centre til at styre opkald mellem medarbejdergrupper.

Realistiske call center modeller må nødvendigvis kunne bruges på systemer med flere betjeningssteder. Med udgangspunkt i dette, udvikles der en approksimation i artiklen *Waiting time dependent multi-server priority queues*, der kan bruges på sådanne systemer. Den bruges til at analysere en opsætning med to kundeklasser og to medarbejdergrupper, hvor der desuden er mulighed for at sende opkald i overløb mellem grupperne efter en regel baseret på en fast grænseværdi på ventetiden. Ydermere demonstreres approksimationens alsidighed ved at anvende den på et system, hvor kunder prioriteres efter en dynamisk regel.

I den sidste artikel, *Optimization of overflow policies in call centers*, analyseres overløbet mellem medarbejdergrupper nærmere. Overløbet optimeres ved hjælp af Markov beslutningsprocesser og det vises, at en forbedring af behandlingen af kunder kan opnås. Desuden påvises det, at overløbsreglen baseret på en fast grænseværdi på ventetiden er fordelagtig i tilfælde, hvor en kundeklasse tildeles høj prioritet, samt når det tilstræbes, at kunder betjenes af de rette medarbejdere, dvs. at de ikke sendes i overløb.

Contents

Acronyms

ACD Automatic Call Distributer
AE Average Excess
ANI Automatic Number Identification
ASA Average Speed of Answer
AWT Acceptable Waiting Time
CSR Customer Service Representative
CTI Computer Technology Integration
CTMC Continuous Time Markov Chain
DNIS Dialed Number Identification Service
FCFS First Come First Served
FCR First Call Resolution
FIL First In Line
IP Internet Protocol
IVR Interactive Voice Response
MDP Markov Decision Process
PABX Private Automatic Branch Exchange
PSTN Public Switched Telephone Network
QED Quality-Efficiency Driven
RPC Right Party Contact
RSF Rostered Staff Factor
SLA Service Level Agreement
SLT Service Level Target
TEA The Erlang Approximation
TSF Telephone Service Factor

CHAPTER 1

Introduction

Call centers are the preferred way for most major companies and public institutions to interact with customers and citizens. The call center industry is by all measures large and still growing. As of 2006, 2.2 million jobs exist within the industry in the US alone (about 1.4% of the entire workforce) with an expected growth of 25% in the period 2006-2016 [18]. Because of the size and labour intensive character of the industry there is a large degree of motivation to understand the dynamics of call centers to ensure proper management and avoid unnecessary use of resources.

The main driving forces behind aggregating customer contact in call centers are *economy of scale* together with the possibility of having a higher degree of consistency in the interaction with customers. A large call center allows for more efficient operation, however, it also induces more complex operational processes. Advanced mathematical methods are required to understand the details of the processes in order to optimize and improve the operations.

Management within the call center industry involves many areas of operations management such as forecasting, scheduling, capacity planning and routing optimization. In [21], call center management is described as a circular iterative process. It starts out setting a desired service level target to aim for. The next element is data collection from which, forecasting can be carried out. After having forecasted the expected load, the required staff and physical structures can be found taking shrinkage (e.g. absence of employees)

into account. Finally the actual scheduling of agent work hours can be settled and the cost of the operation calculated. The process can then start over and the service level can be adjusted to ensure that excessive costs are not spent on a too high service level or vice versa.

The above description of call center management is a very general presentation. The elements included in the planning process are described in further detail in Chapter 2-6. Especially the advent of the highly computerized setup in modern call centers contributes to more complicated systems. The computerized setup allows a high degree of customization of how calls are handled. This means that customers can be treated in different ways and the agents can be divided into groups according to their skills. Such arrangements are referred to as multi-skill call centers. Within such a setting, the disciplines of call routing and prioritization become important.

The iterative planning process may be fine for smaller adjustments. However, for larger changes to the infrastructure of call centers it is essential to understand what the consequences of these changes will be. By modelling the call center setup, the effect of changes can be assessed without the risk of impairing the system during the process.

This thesis centers around routing and handling of calls and is arranged as follows. A general introduction to the call center world is given in Chapter 2. This includes a description of the physical structure of call centers and the central concept of service levels.

Chapter 3 deals with analysis of traffic data. The characteristics of the traffic to inbound call centers are described together with different aspects that may affect it. The importance of data collection for analysis of e.g. waiting time, service time, and abandonment distributions is treated. Finally the use of historical data for forecasting purposes is described.

Key issues of capacity planning in call centers are treated in Chapter 4. This basically involves having the right number of agents and the right use of these to accommodate incoming calls in the desired way. All the different levels of the planning hierarchy are treated, this ranges from resource acquisition, such as designing the physical surroundings and the hiring and firing of call center staff to the more detailed management issues. These include staffing and rostering of agents, i.e. determining how many agents should be at work at a given time and how the work schedules of the individual agents should be designed to best correspond to the need while adhering to rules regarding work hours. Also the complex issue of call routing in multi-skill call centers is discussed.

In Chapter 5, routing policies based on the waiting time of the first customer in line are treated. The special case characterized by fixed thresholds is dealt

with in detail. This policy is used widely in multi-skill call centers for load balancing between different agent groups. The discussion in this chapter is centered around the three papers presented in appendices A, B, and C, which all deal with the analysis of thresholds.

The use of simulation for modelling call centers is treated in Chapter 6. This includes a discussion of the benefits and disadvantages of using simulations as compared to analytical models or approximations. At the end of the chapter, the possibilities of using simulation for modelling call center performance are illustrated through a concrete example.

Finally in the last chapter of the summary report, a short discussion of the contribution of the thesis is given. The main contribution of the thesis is presented in the appendices in form of three journal papers.

In Appendix A a new approach to modelling queueing systems is presented. This involves using the waiting time of the first customer in line as the primary variable, which the analysis is based on. The approach is in many ways similar to the one used in [10], where the workload is considered. The new approach is especially convenient when queueing systems with operations that somehow depend on the experienced waiting time of customers are considered. Two such systems are analyzed in Appendix A, one where the service rate of a server can be adjusted according to the waiting time of the customer first in line. The other analyzed system involves two servers, of which one is only allowed to take customers who have waited a certain fixed amount of time. The latter system is based on a common rule used in call centers for skills-based routing, where calls are only allowed to overflow between different agent groups after having waited a fixed amount of time. The analysis in Appendix A is elaborated further on in Appendix A.1 where an additional mathematical investigation regarding the independence of a linear equation system from the paper is given.

A more application oriented approach is taken in the work presented in Appendix B. Here an approximation, based on the same approach of using the waiting time of the customer first in line as the primary variable, is developed. The approximation is used to model a realistic call center multi-server setup with two customer classes and two agent groups and overflow from one class to the other. Furthermore, the general nature of the approximation is illustrated by modelling a dynamic priority system. In Appendix B.1 the validity of the approximation in its simplest form is proven.

The last paper, presented in Appendix C, deals with optimization of overflow. Again, the same approach of analyzing the waiting time of the first customer in line is taken. The appropriateness of using the fixed threshold policy is also examined.

CHAPTER 2

Call Center Basics

The fundamental aspects of call centers and call center management are treated in this chapter. This includes the background and motivation for using call centers in Section 2.1 and the physical structure in Section 2.2. The use of different service measures is discussed in Section 2.3.

2.1 Background

The term *call center* covers a large variety of instances. Another often seen term, is *contact center*, where the general consensus is that a call center only communicates through phone calls whereas contact centers have multiple communication channels such as email or chat beside phone calls. The term call center may also be avoided in some cases due to it having a somewhat negative connotation. This may especially be the case for call centers with highly skilled employees who want to differentiate their service from simpler ones often provided by students working part time. The term call center is used throughout this thesis as mainly the discipline of handling calls is considered. See e.g. [15] for a treatment of contact centers with traffic composed of mixed mail and phone calls.

The main purpose of gathering customer contact in one central place is economy of scale, a large central unit is more efficient than multiple smaller units. The simplest explanation for this is that the risk of one customer having

to wait at one place while capacity is available another place is eliminated. This is treated further in Section 4.2.

Call centers are divided into inbound and outbound depending on who initiate the calls. The term inbound call center is used for the case where customers initiate calls, typically to obtain a service. Outbound refers to call centers where agents call customers. The latter would typically be used to sell a product or conduct a survey. Combinations of the two are of course also possible. Inbound call centers are the most interesting from a modelling point of view as this is where most of the challenges are, due to the stochastic nature of incoming calls. Outbound call centers can to a large extent decide when calls take place, whereas there is a large element of uncertainty associated with inbound call centers. For this reason, this thesis focuses solely on inbound call centers. Optimization of a combination of in and outgoing traffic is treated in [11].

People can have many reasons for communicating with call centers. The most obvious is a person calling a company he does business with, such as a bank or a support hotline. For outbound call centers it can even be involuntarily by salespersons calling and trying to sell services or goods, often at the most inappropriate of times. This may very well be one of the sources of the somewhat bad reputation call centers can have. Emergency services, such as 112 and 911 in the European Union and the United States respectively, also fall under the category call centers. We will refer to people communicating with a call center as customers no matter the reason.

There is an abundance of general introductions to call centers and call center management. Highly recommended for a lighter approach to call center management is [21] whereas [25] gives a more mathematical introduction including numerous references.

2.2 Call Center Structure

The main resource of call centers is the employees responsible for the communication with customers. The used terms for these employees are agents or Customer Service Representatives (CSRs). The fact that salaries account for 60-80% of total operating costs [50], [4] makes this by far the most important subject for optimization.

Operations in modern call centers are built around an extensively computerized setup as depicted in Figure 2.1. Call centers typically have their own telephone exchange in the form of a Private Automatic Branch Exchange

(PABX) that connects phones internally and to the Public Switched Telephone Network (PSTN) through trunk lines. For further details of the underlying telecom technology see e.g. [53].

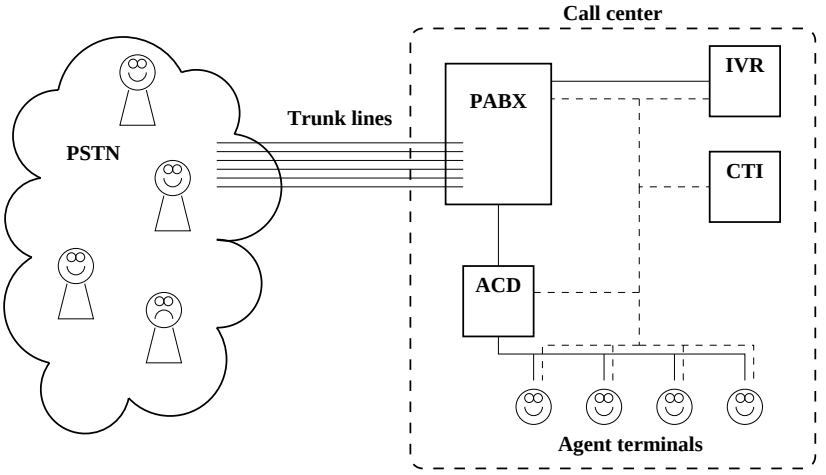


Figure 2.1: The physical structure of a modern call center. Solid lines represent voice connections and dotted lines data connections.

The effect of limited computer system processing power is analyzed in [3] by treating the agent pool as a processor sharing unit. This case should really be avoided though, as payroll costs are generally dominant in call centers. One could maybe imagine cases in parts of the world where salaries are very low, where this would be relevant.

A central piece of equipment is the Automatic Call Distributer (ACD) which distributes incoming calls and collects call statistics. This makes it important for planning purposes. The distribution of calls is based on algorithms that can be designed to accommodate calls in desired ways. This topic is discussed further in Chapter 4.

Interactive Voice Response (IVR) is the technical term for the equipment that gives welcome messages, plays music, and informs about the caller's place in queue, if these services are implemented. The IVR also allows users to interact with the call center using the keypad of the phone. This includes letting the customer choose a desired service after hearing a message such as

Press 1 for sales, press 2 for technical support... etc. The IVR also handles automatic phone services where customers can be served or obtain information without actually talking to an agent. Examples of such services is activation of credit cards using a secret code sent by mail, checking the balance of a bank account, and even filing a tax return.

Another used practice, to allow customers to choose a service, is to use different phone numbers for different services or indeed prioritize customers by giving important customers another number to dial and thus get ahead in line. The call center can then identify the called number using Dialed Number Identification Service (DNIS) and treat the call accordingly.

Many companies employ a set of "lines of defence" as illustrated in Figure 2.2. The goal of these are to satisfy whatever needs customers may have without reaching an actual agent in the call center. The first line of defence can be a web page, where customers can find the information or service they need, home or internet banking is an example of this. The next step can be an automatic phone service where customers can get the information or do simple actions using the numeric keypad of their phone. Only if customers are not able to take care of their business in the first lines of defence, they get through to an agent. The service provided by agents may again be divided into multiple layers. The first level could be less skilled (cheaper) agents capable of handling simple requests. The second level could then be experts taking care of the more advanced subjects. The purpose of this is to reduce the time agents must spend talking to customers, and in this way decrease the required number of agents and thus the expenses.

A way to reduce the duration of calls is by Computer Technology Integration (CTI), which enables automatic retrieval of customer records when calls are put through. This can be implemented by letting customers identify themselves while interacting with the IVR or directly using Automatic Number Identification (ANI) thus removing the need for agents to spend time entering e.g. account or social security numbers given verbally. Instead the agents can have all information about the customer ready on their screen as soon as the call is put through.

In recent years Internet Protocol (IP)-telephony has become more common, also within the call center world. IP-telephony allows for a larger degree of flexibility and should cut down on infrastructure costs. Cost reduction can be obtained through higher bandwidth efficiency and by having a unified communication system for voice and data including fax, e-mail, access to the Internet, etc.

The computerized and highly customizable setup of modern call centers

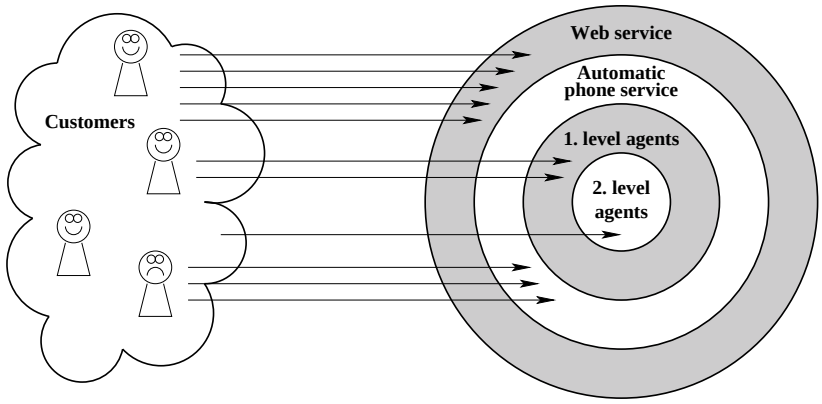


Figure 2.2: Illustration of the typical lines of defences a company uses to decrease the number of calls (and thus the cost) to a call center. The call center is represented by the inner three circles and agents the inner two.

allows for a virtual single call center to be physically spread out over different sites. The motivation for this can be to draw from workforce pools in different regions of a country or different countries for multi-national companies. This may even be used to exploit different time zones by having a call center in e.g. Australia, Europe and America thus allowing each of these to be open only during work hours while providing a 24-hour service. Another possibility is to allow agents to work from home and still be a part of the virtual call center. Besides the obvious personal benefits to the employees, having people work from home may benefit the employer by reducing the number of office seats needed. Often full-time employees will prefer continuous full day shifts, however, working from home may convince agents to take shorter shifts during peak periods, thus increasing flexibility and making the assignment of work shifts easier.

2.3 Service Measures

Providing a service is the fundamental objective of inbound call centers. The concept quality of service involves different aspects. Some of these are quantitative and easy to measure, others are more qualitative and thus harder to

measure.

Call centers can be characterized by how they are operated, distinguished by different regimes. Some call centers may have a very low agent utilization meaning that almost all calls are answered immediately, this is referred to as the quality driven regime. At the other extreme we have the efficiency driven regime where almost all customers have to wait before being served, here the agent utilization will be close to 100%. In between, and where most call centers operate, is the rationalized regime where agent utilization is high (maybe $90 - 95\%$), but many calls may still be answered immediately. This is made possible by large call centers enjoying the benefits of economy of scale. The different regimes influence how system performance can be approximated. The important rationalized regime was first thoroughly analyzed in [34] resulting in its alternative designation, the Halfin-Whitt regime. Sometimes it is also referred to as the Quality-Efficiency Driven (QED) regime.

Service Levels

The quantitative service measures typically involve how long customers have to wait before the call is answered by an agent. The simplest measure of this is Average Speed of Answer (ASA), which is, as the term implies, just the mean of the time customers wait in queue. The measure used by itself is lacking in detail, as no information about the waiting time distribution is given other than its mean.

In in-house call centers the upper management typically sets a Service Level (SL) target for the operations manager to meet. For outsourced call centers the call center operator and the company enter a contract in the form of a Service Level Agreement (SLA). Service levels are common in the simple form of a specified percentage of calls that should be answered within a certain time, referred to as Acceptable Waiting Time (AWT) or Service Level Target (SLT). This is written as e.g. 80/20 which means 80% of calls should be answered within 20s. This service measure is referred to as Telephone Service Factor (TSF).

Lost or abandoned calls are also normally considered. This measure would typically be strongly correlated with the service levels. Customers may even be encouraged to abandon and call back while they are in queue if the system is overloaded in order to improve service levels.

The TSF used in call centers varies greatly; 90/20, 85/15 or 90/15 would typically be regarded as high service levels used in competitive businesses where customers may take their business elsewhere if service is unsatisfactory. For

emergency services the demands may be even higher. A TSF of 80/20 is the classic middle of the road choice often used in banks, insurance companies, and travel agencies. In the lower end, service levels such as 80/300 may be used by e.g. soft- and hardware support centers and governmental offices where you can't really take your business elsewhere [21].

Service levels have to be approached with some caution. TSF used by itself actually motivates tossing customers out of the queue as soon as the time they have spent in the queue exceeds the threshold of the TSF. This approach will increase the chance of serving other customers within the threshold. Another similar approach would be to only serve customers which have exceeded the threshold if no customers having waited less than the threshold are present in the queue. In order to account for some of these discrepancies a new service measure is suggested in [49]. The suggested service measure is called the Average Excess (AE) and it is, as the name suggests, the average time, which waiting exceeds SLT.

It has been suggested to introduce a third term to TSF representing the percentage of intervals where a percentage of calls are answered within a time limit. E.g. 90/80/20 would mean that in 90% of time intervals 80% of calls should be answered within 20s. The length of the time interval should also be given.

The widespread use of TSF probably originates in how easily it is interpreted. Despite this, less mathematically gifted management may still require the impossible of the operations management team, such as a 100% service level with a low SLT for their most priced customers. It may seem appropriate to guarantee the most valued customers to be answered quickly, however there must be room for exceptions as 100% service levels in principle require an agent sitting ready for each customer.

There is some discrepancy about how to calculate the performance measure TSF from data. This is discussed in [21] and involves how abandonments are dealt with. Sometimes abandoned calls are disregarded altogether, and TSF is calculated as the number of calls answered before the SLT divided by the total number of answered calls. Another option is to divide by the number of offered calls which will degrade the TSF if many calls abandon. Sometimes only calls that abandoned before SLT are included. As such there is no unique right way to do it as long as it is clarified how it is done. However, the method of dividing the sum of answered and abandoned calls within SLT by offered calls is recommended in [21].

Even taking the different ways to calculate TSF into account there is still room for more interpretations. The time period over which the TSF is mea-

sured, can have a major influence on the resulting values. If TSF is measured over a short period, the percentage of calls answered within the time limit will vary much more than if longer periods are used. During a 24-hour period the required number of calls may be answered within the time limit, however if the same day is split into hourly intervals the TSF may be exceeded during some of the intervals.

Customer Satisfaction

The qualitative part of call center service is harder to measure. This includes the customer's impression of the conversation, such as politeness and competence of the agent. Agent competence plays a major factor in First Call Resolution (FCR) which is important because it affects the number of calls a call center receives. Customers are likely to call back if their intent is not met the first time.

Many call centers use the possibility of having supervisors or colleagues sitting next to them and listen in on the calls. In this way agents can receive feedback on their performance and be coached into giving better service by the listener-in. Some call centers also have supervisors listening in on calls to assess performance without agents knowing it. This may not be a good practice from a work environment point of view, as agents may feel monitored.

The usual suspects when it comes to what determines service quality and customer satisfaction are ASA, abandonment rate, blocked calls, TSF, FCR, and talk time. The first three can be expected to be negatively correlated and the latter three positively correlated to customer satisfaction.

A study in [23] shows a surprising lack of correlation between the usual performance measures (ASA, abandonment rate, talk time, etc.) and customer satisfaction in banking/financial call centers, however, no explanation of what determines satisfaction is given. This essentially highlights the issue of the assumption that satisfaction is correlated to what can easily be measured rather than harder to measure factors such as agent kindness, etc. An interesting result of [23] is that correlation between the usual performance measures is much larger for call centers within other industries.

CHAPTER 3

Traffic Analysis

The nature of call traffic to inbound call centers is highly stochastic. Also the duration of calls and the caller behavior with regard to patience and the tendency to redial, constitute interesting research topics which are discussed in this chapter.

3.1 Traffic Characteristics

The traffic to inbound call centers varies greatly with time. This variation can be seen as a function of the time of day, day of the week, day of the month, month of the year, and even the year.

The most obvious variation is typically seen intra-day as calls depend on the work hours and daily routines of people. The business hours of a call center may also influence the pattern of arriving calls. This could be in the form of a peak just after opening time. A pronounced variation over the week will also often be seen, mostly with a distinction between weekdays and weekends. Variation over the month may especially be seen at the start and end of the month as many bills have to be settled around this time. Also people may have a tendency to spend more money on shopping on and just after payday, resulting in an increased call load to sales call centers. An intra-yearly variation will typically result from people going on vacation during the summer, Christmas, etc. Inter-yearly variation can be assumed to come from a general trend in call

loads originating in a changing number of customers or people's disposition to use the call center. The possibility of identifying a general trend motivates saving records of call volumes for a long time.

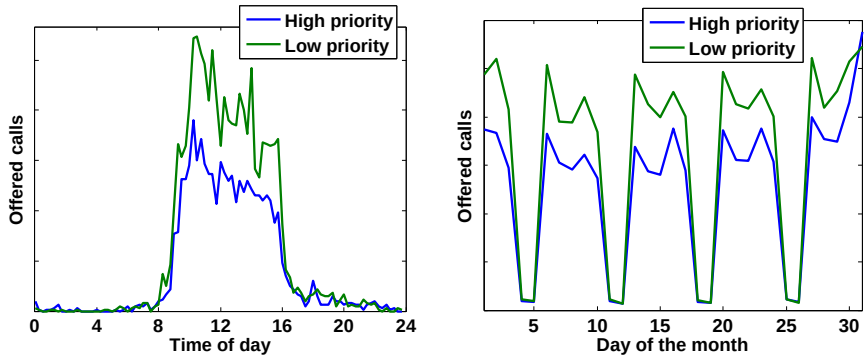


Figure 3.1: Calls to an inbound call center during 15 minute intervals over a day and daily intervals over a month. The number of calls are depicted on a linear axis without a scale as the intention is only to illustrate the patterns.

The number of calls a call center receives is referred to as offered calls. This can be considered over any given time interval, such as 15 minute or daily intervals and includes all calls including both those whom are eventually served and those whom abandon. Figure 3.1 shows an example of the pattern of offered calls over a day and over a month for low and high priority calls. The daily variation is clearly seen with the highest load during work hours and in the morning in particular. The weekly variation is also clear with fewest calls in the weekends and most on Mondays. The previously mentioned increase around the turn of a month seems especially pronounced for the high priority customers. Also the offered calls of high and low priority customers appear highly correlated. Observations such as these should all be accounted for when designing forecasting models as described in Section 3.3.

Calls to an inbound call center are often assumed to arrive according to a Poisson process (i.e. times between calls are independent and exponentially distributed) with constant intensity within short time intervals. The rate at which calls arrive is generally denoted by λ. The reasoning behind this is that the total number of customers is assumed to be infinite, thus the intensity of

arriving calls does not depend on the active number of calls. This assumption is appropriate as the customer base of call centers is normally quite large.

If the average service time of calls is also given, it makes sense to consider the offered load. The offered load is measured in erlang (E), a dimensionless unit, given by multiplying the mean arrival rate with the average service time as

$$A = \lambda\mu^{-1},$$

where μ^{-1} is the average service time and A the offered load.

Redials due to calls not being resolved during the first contact will violate the Poisson assumption as the arriving calls will be dependent on the service level and abandonment rate. This means that the way calls are served, influences the offered traffic, contrary to what is normally assumed in most models, i.e. the offered traffic being independent of the service quality [37].

3.2 Data Collection

Collection of data in call centers is essential for planning purposes and for analyzing performance. Large quantities of data are normally logged as data storage capacity is relatively cheap. This includes number of calls, duration of calls, waiting times, abandoned calls, etc.

A widely used practice is only to log aggregated call data as this results in much less data compared to logging of individual calls. Aggregated data would typically be the number of offered, abandoned, and answered calls during a 15 minute period. Waiting time distributions can also be logged in an aggregated form as number of calls served after having waited between 0 and 2 seconds, 2 and 4 seconds, again during 15 minute intervals. In this way a lot of information is discarded compared to logging call-by-call data.

The more detailed call-by-call data is e.g. required to analyze the arrival process in order to investigate the assumption of it being Poisson in short time intervals. Also call-by-call data could be used to estimate an arrival function, $\lambda(t)$, which would be interesting for models dealing with non-stationary traffic as discussed in Chapter 4. The same goes for detailed analysis of the waiting time distributions, both for those who are eventually served and those who abandon. Aggregated data should be adequate for most forecasting purposes though.

Another topic for which aggregated data cannot be used to examine is first call resolution. Call-by-call data should be more appropriate for this as the

same caller calling multiple times within a short time may very well indicate that his intent was not resolved by the first contact.

The literature dealing with statistical analysis of center data is not as abundant as it could be. This is to some extent probably due to the lack of reliable call-by-call data. Exceptions are [16] and [17] where the same call-by-call data from an Israeli call center is analyzed.

Service time

For most analytical work, service times are assumed to be exponentially distributed. This is often done for practical reasons as exponentially distributed holding times are easiest to deal with analytically even though it may not always be the best representation of reality. It also means that only the mean of the service times needs to be used for modelling. The rate of which calls are handled by agents is normally denoted μ, i.e. the mean service time is $1/\mu$.

A detailed analysis of call center service times can be found in [17], where the service times are shown to be very close to lognormally distributed. The mean service time was also shown to vary greatly over the 24-hour day with the longest service times coinciding with the highest offered traffic in the late morning and afternoon. This was especially the case for regular calls to the bank, but less so for internet support calls which had a more stable service time.

The problem of short service is also discussed in [17]. This involves agents hanging up on customers as soon as they are put through in order to increase the number of calls the agent handles. This obviously decreases the mean handling time, which is a part of how agents are evaluated. This issue stresses the importance of not relying on too simple measures as these may be exploited by the more scrupulous individuals. By considering the entire service time distribution this issue was easily identified and appropriate measures could be taken.

Further analysis can be found in e.g. [13] which treats the holding (service) time of telephone calls as a mixture of different components. Also the human perception of time is considered.

As calls constitute a dialogue, service time of a call is obviously dependent on both the caller and the agent. However, in most multi-skill models the service time is bound to either the agent pool or the customer class, often out of convenience.

Customer Patience

Customers are in simpler models, such as the Erlang-C model treated in Section 4.2, assumed to be infinitely patient. This is however not realistic in a call center setting as customers get tired or annoyed and abandon the queue. Ignoring abandonments can actually lead to overstaffing as the carried load no longer equals the offered load.

Some call centers inform customers about their position in queue while they are waiting. This may influence their patience in either way. Detailed studies of this are lacking though. What is seen from the waiting time distributions of customers who abandon, is that announcements cause some people to abandon, whether the announcements give the position in queue or not. An example of this is seen in Figure 3.2 where an announcement without information about position in queue or expected waiting time results in increased abandonment. Announcements may cause other people to wait longer, but this is hard to quantify.

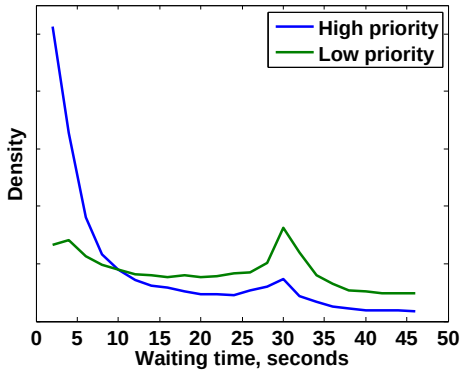


Figure 3.2: Waiting time distribution before abandonment. The peak at 30s coincides with a "please hold"-message.

If high and low priority customers are routed to a common agent pool, it may be inappropriate to inform customers about their position in queue. In this case a low priority customer may experience being pushed further back in queue by high priority customers arriving while the low priority customer is waiting. Such an event would be very aggravating for the customer being

pushed back and probably not something call centers would expose even low priority customers to.

Another option is to give an estimate of the remaining waiting time. This can obviously never be more than an estimate, however it actually gives more valuable information to the customer than just giving the position in queue. For example being informed that you are number eight in queue may mean you have to wait for a very long time if only one agent handles calls or a very short time if hundreds of agents are working. Prediction of waiting times is examined in [74] where information such as the number of customers in queue and elapsed waiting time for non-exponential service times are used to improve estimates.

An interesting note is that in an $M/M/n$ system with a First Come First Served (FCFS) policy and visible queue the waiting time a customer can expect to experience is gamma distributed, but if the queue is invisible then the waiting time is exponentially distributed [37]. This is due to the fact that the expected number of waiting customers is geometrically distributed and a set of gamma distributions with shape parameters weighted by a geometric distribution does indeed result in an exponential distribution.

An empirical analysis of abandonments is given in [78]. It is shown that experienced callers adapt their patience according to system performance even if information about queue length is not given, they simply learn that at certain times they risk waiting a long time if not being served immediately.

What is often referred to as the Erlang-A model extends the Erlang-C model by including abandonments. Analysis of models including abandonments was first carried out in [60]. Erlang-A can be seen as a generalization of the Erlang models, with Erlang-B and Erlang-C being the two extremes with zero and infinite patience of the customers.

Analysis of customer patience requires inference analysis because the patience of those served is not known and is thus considered censored data. Empirical data of customer patience is examined in [16] and [17] and shown to have a nearly constant hazard rate, i.e. the patience is nearly exponentially distributed. The main deviation from this comes from messages being announced, stating that the customer is in queue. These messages seem to make customers abandon the queue, intentionally or not. Customer patience is further analyzed in [27] where a method based on boot strapping to determine confidence intervals is presented.

3.3 Forecasting

The discipline of predicting the number of incoming calls to a call center is referred to as forecasting. The importance of thorough forecasting cannot be overestimated. Unprecise predictions of the call load lead to either over- or understaffing which in turn results in unnecessary pay costs or unsatisfactory service levels respectively.

In general, a forecast will be more precise, the shorter the time horizon. Forecasting with a horizon of a month or more is used to determine the required staffing levels and from those, schedule the work hours of agents. Forecasting with a shorter horizon of, say a few days, could be used to call in extra staff if needed.

It is often assumed that calls arrive according to a time-inhomogeneous Poisson process. The inhomogeneity stems from the fact that the probability of a customer making a call varies during the day, week, month and year. The rate of arrivals is normally denoted λ for time-homogeneous systems and λ_t or $\lambda(t)$ for time-inhomogeneous systems.

The most common approach to forecasting call volumes is a simple top-down approach [25]. This approach starts with an estimate of the number of calls during a month or even a year from historical data. From this, weekly, daily and typically half-hourly estimates are made. As an example, assume that a call center receives $1'000'000$ calls in a year; from historical data it is known that 10% of calls are made in May, of those, 20% of calls fall in the first week of the month, 25% of calls are made on Mondays and 5% are made between 10.00 and 10.30, then the estimated number of calls in the half-hourly interval would be $1000000 \cdot 0.1 \cdot 0.2 \cdot .25 \cdot 0.05 = 250$. Due to the simplicity of this approach, one might expect that better results would be easily obtainable. Certain days such as national holidays will not follow this pattern and are often dealt with from the past experience of an operations manager.

In [69], a number of different approaches to forecasting are examined using univariate time series methods. The result of this work indicates that a simple approach referred to as *Seasonal mean* is hard to beat when the horizon of the predictions is more than a few days. *Seasonal mean* is just a moving average over the same half-hour of the week of previous weeks. For shorter forecasts with a horizon of 1-2 days another method, *Exponential Smoothing for Double Seasonality*, was shown to perform better. This method assigns more weight to the most recent observations and takes both intra-day and intra-week seasonal cycles into account.

Also of importance when forecasting is to take the expected service time

into account. It has been observed that the average service time can increase during evenings, effectively making the call load higher, see [21] and [17]. Also a (positive) correlation between the call volume and mean service time has been observed as agents may become tired and less effective when they are stressed [16].

In [65] detection of outliers, data visualization, and determination of significant features such as day of the week effect from noisy data are treated using singular value decomposition. This is used to develop a forecasting model and used for a test case based on data from an American financial company. The outliers are found to nicely correspond to holidays or days with system errors and a day of the week dependence is identified, both contributing to improving the forecasts.

Computation of prediction intervals for the arrival rate is treated in [44]. By using Poisson mixtures to deal with overdispersion, the arrival rate is treated as a random variable, i.e. the arrival rate λ is drawn from one distribution and the number of calls in an interval is then drawn from a Poisson distribution with parameter λ. This is necessary as the variance of observed data is often higher than the mean, these should be even under the Poisson assumption. Call center performance is finally determined using the Erlang-C formula.

CHAPTER 4

Call Center Capacity Planning

The issue of determining the number of servers required to provide customers with a desired service level can be very complicated due to the many factors involved. This topic is treated in the following sections.

4.1 Resource Acquisition

A part of the physical structure to be dimensioned is the number of telephone trunk lines. All active calls need a trunk line, both those queued, those using the IVR, and obviously those which have been put through to an agent. In most cases call centers ensure that there are simply enough trunk lines for almost any scenario as the required hardware is relatively cheap compared to agent wages. The number of required trunk lines can be determined by using the Erlang-B formula [37] as calls are blocked if no line is available. This is further treated in Section 4.2

The hiring process of agents also falls within resource acquisition. Depending on the required skill of agents, hiring and training a new agent may take days, months or even years. This obviously calls for some long term planning taking into account the expected growth of business, i.e. new customers or future acquisitions of other companies. Also people leaving for other jobs and

retiring must be considered. The hiring problem is not unique to the call center industry and is well described in the literature such as in [31].

Shrinkage

The number of agents required to man the phones at any given time does not equal the number of agents that needs to be scheduled for the given time. The actual number of available agents is decreased due to many causes such as people calling in sick, needing breaks to go to the toilet and get coffee, lunch breaks, meetings, coaching, education, vacation, and more. Some factors can be controlled to some extent as discussed in [1] while other may be more problematic.

Accounting for shrinkage is normally done by fairly simple methods such as described in [21] and [20]. The procedure is to use Rostered Staff Factor (RSF) (also referred to as overlay) as a factor which describes the required overstaffing needed to account for shrinkage. Not much work in the literature takes absenteeism into account for general models but [76] treats the number of active agents, given the number of agents on job, as a random variable.

4.2 Staffing using Queueing Models

The fundamental aspect of staffing is to determine the minimum number of agents required to satisfy service levels. Obviously overstaffing leads to increased costs and understaffing results in poorer than desired service. This means that staffing is a key element in call center management, i.e. where great savings can be made. In this way, a good result for an operations manager is not to have as many calls as possible answered within the TSF, but rather just the required percentage.

The many different issues related to the flow of calls through the call center are illustrated in Figure 4.1. Most models only take some parts of the process into account. This will often be due to necessity as models quickly become very complicated, however much insight can also be gained from simplified models.

The discipline of determining the required number of servers to offer incoming traffic a given service level is the most classic within queueing theory. It was pioneered by Agner Krarup Erlang while he worked at the Copenhagen Telephone Company with the aim of determining the number of required circuits to ensure the desired telephone service. He also worked on determining how many

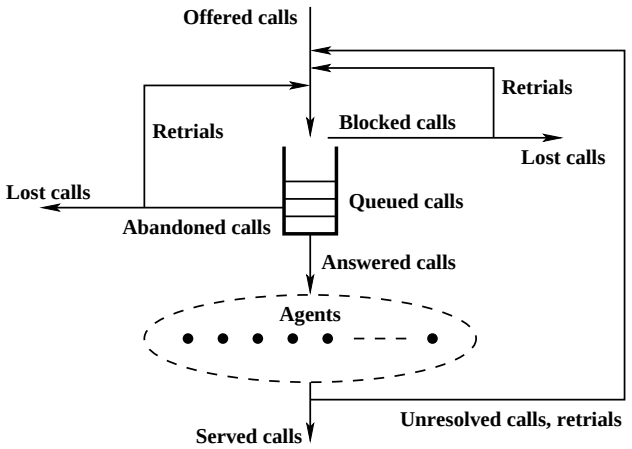


Figure 4.1: Call center process diagram.

telephone operators should be working at cord boards used to switch telephone calls in the early days of telephony. In many ways this can be seen as the ancestral origin of modern call centers or indeed the mathematical aspects of call center management. Erlang's work has proved extremely long-lasting and his formulae are still widely used within the call center and telecommunication worlds.

Erlang-C

The most used way of determining the number of agents needed, is by the Erlang-C formula. The model behind the Erlang-C formula is $M/M/n$, Markovian arrivals and service completions (time between arrivals and service durations are exponentially distributed) and n servers with infinite waiting positions and infinitely patient customers. The formula gives the relation between the delay probability (the probability of not being served immediately, $\mathbb{P}(W > 0)$) and the number of servers (n), arrival rate (λ) and service rate (μ). It is worth noting that only the ratio between λ and μ matters, thus the offered traffic is introduced as $A = \lambda/\mu$. See for [37] for details and references on the original

work of Erlang.

$$\mathbb{P}\left(W > 0\right) = \frac{\dfrac{A^n}{n!}\dfrac{n}{n-A}}{\displaystyle\sum_{i=0}^{n-1}\frac{A^i}{i!} + \frac{A^n}{n!}\frac{n}{n-A}}.$$

Assuming that calls are handled on a first come, first served (FCFS) basis, then the waiting time distribution can be found as

$$\mathbb{P}(W \leq t) = 1 - \mathbb{P}\left(W > 0\right) \cdot e^{-(n\mu-\lambda)t}, \qquad n > A,\ t > 0.$$

The use of the Erlang-C formula is fairly straightforward as it only requires estimation of the arrival rate and mean service time, which should be available from forecasts based on historical data. However, the Erlang-C model has some shortcomings. A major one being that it does not take customer abandonments into account. This means that if the offered traffic exceeds n, then the queue becomes unstable and grows towards infinity.

In order to have a stable system, i.e. the queue will not grow infinitely, the number of servers must be greater than the offered traffic for an $M/M/n$ system. The number of servers beyond the offered traffic is referred to as safety staffing and a well proven rule in this context is the square root safety staffing principle. In words, this rule dictates that in order to have approximately the same service level, the number of additional servers beyond the offered traffic should grow as the square root of the offered traffic. Put formally it becomes

$$n \approx A + \beta\sqrt{A},$$

where n is the number of servers, A the offered traffic and β a coefficient related to the service level. The rule applies best to heavily loaded large systems. The principle behind square root staffing has been known for long, as it is fundamentally based on the central limit theorem. In fact Erlang used the principle, but it was formalized in the call center setting in [14]. The quality of the approximation behind the square root principle is examined in [41] and shown to be very good even for moderate sized systems as long as abandonments are ignored.

The square root safety staffing rule nicely illustrates the economy of scale principle, one of the main driving factors of call centers. A simple example of this would be to consider a call center offered a load of 100 E compared to one offered 400 E. To obtain approximately the same service level (assume $\beta = 1$),

the smaller call center would require a safety staff of 10 agents compared to 20 for the four times larger call center, i.e. only twice as many safety staff despite a fourfold larger traffic. Put in another way the utilization of agents increases with the size of call centers, given the same service level as described in [73].

Erlang-B

Also well known is the Erlang-B formula, which is based on a rejection model. This means that customers that are not served immediately are blocked and lost. Arrivals and service completions are assumed Markovian as for the Erlang-C formula. The Erlang-B formula gives the probability of being blocked, B [37]:

$$B = \frac{\frac{A^n}{n!}}{\sum_{i=0}^{n} \frac{A^i}{i!}}.$$

The Erlang-B formula is normally used for trunk capacity calculations [21] in the call center world, that is how many phone lines should go into the call center.

Beyond Erlang

There is an abundance of more advanced queueing models in the literature. As the assumptions behind the Erlang-C model differ from real scenarios, there should be room for better models. The more advanced models typically take abandonments, different distributions of service time or patience, varying load, etc. into account. Some of the literature on these models is discussed here.

A simple extension to the Erlang-C model is to limit the number of queueing positions which in practice is also the case in real call centers due to the limited number of trunk lines. However, as the number of trunk lines is often dimensioned generously, it may very well not be the most important extension of the Erlang-C model to include, when considering call center cases. The usual notation for this kind of system is $M/M/n/k$ (given Markovian arrival and service processes), where the k refers to the number of queueing positions. Some discrepancy exists in the literature whether to include the n servers in the k queueing positions or not, something to be mindful of.

A more important extension is the inclusion of abandonments in the model. Neither the Erlang-B nor Erlang-C formula describes the dynamics of a call

center very well as people tend to have neither zero nor infinite patience. In this way the Erlang formulas can be seen as two extremes. Many models trying to fit in between these have been introduced. A simple way of introducing patience is to assume customers have exponential patience, thus keeping the Markovian property as done in [60]. This has later been referred to as the Erlang-A model (A for abandonment) in [26] where the importance of taking abandonments into account when modelling call centers is examined. It is shown that ignoring abandonments can result in very significant overstaffing. The use of the square root safety staffing principle for the Erlang-A case is examined in [77] and the addition of a correcting term is suggested.

The assumption of exponential service times in call centers does not fit well with empirical data as discussed in Section 3.2. This means that underlying assumptions of the analytically attractive Erlang-C, $M/M/n$ model are violated. This calls for the use of the $M/G/n$ model (G referring to general service times) which is less analytically attractive. It is known that the performance of such a queueing system depends on the relation between the variation and the mean of the service time distribution. This relation is typically represented by the coefficient of variation given by the standard deviation divided by the mean or the peakedness given by the variance divided by the mean. This relation is described in [50], where a general introduction to call center queueing models is given. The relation is examined thoroughly in [73] and it is shown that from a performance perspective it is advantageous if the arrival process and service times are of low variability.

As it is generally agreed upon that the service times are not exponential, many models taking this into account have been developed. Such a model is presented in [68] where a general service time distribution is approximated by mixed Erlang distributions (a subclass of phase-type distributions). The approximation takes abandonments, redials, and varying arrival intensities into account.

In [72] it is claimed that an $M/GI/s/r+GI$ model is the most appropriate for call center modelling, where the two GI's refer to general independent and identically distributed service times and abandonments. The reasoning behind this is that the service time and abandonment distributions have been shown to be non-exponential as discussed in Section 3.2. The "correct" model is approximated by an $M/M/s/r + M(n)$ model, where $M(n)$ refers to a state dependent abandonment rate. The use of exponential service times is justified by a claim that service times mostly influence system performance through the mean of the distribution, which is somewhat contradictive to the results in [73].

Another implementation taking customer impatience into account is pre-

sented in [54]. Here workload dependent balking (customers leaving immediately before entering the queue) is used to emulate customers being told an expected waiting time upon calling and hanging up based on this. Phase-type distributions are used for the service time resulting in an $M/PH/1$ model, which is interesting as phase-type distributions can approximate log-normal distributions quite well.

Time-varying Demand

As described in Section 3.1, the arrival rate of calls is far from constant. The simplest approach of assuming the arrival rate to be constant at all times has, not surprisingly, been shown to be a poor approach in almost all cases [29]. This approach is often called the simple stationary approximation and is only really useful in cases where the arrival rate fluctuates quickly compared to the service rate without a general trend. Most often the call arrival process is assumed to follow a Poisson process with constant rate, which may be adequate when a short time interval is considered. This approach of having individual (constant) arrival rates for each interval works quite well for most cases [28] and is often referred to as a point wise stationary approximation. However, in cases with large abrupt changes of the arrival rate it may not perform that well. Such abrupt changes could occur just after a TV commercial promoting a product being sold by the call center or just after the call center opens. The latter is obviously not an issue for 24-hour operations, but could e.g. be relevant for call centers selling tickets for concerts that are being put on sale at certain times. The results in [28] are based on arrival rates following a sinusoidal pattern, but should be applicable to general arrival patterns.

Variations in call loads may lead to a build up of waiting customers in one time interval, e.g. 15 minute interval, which is then carried over to the following intervals. This could be the case during the absolute peak time of the day as the call center may very well be slightly understaffed during this time, because it may not be possible to make rosters that exactly follow the traffic pattern as most agents work full day shifts. This understaffing in one period may lead to further understaffing in the following time intervals as both the carried over calls and the otherwise anticipated calls need to be served.

Fluid models are a useful tool for modelling overload situations. The principle behind fluid models is to consider both arriving calls and served calls as a continuous stream of work instead of individual entities. Fluid models of call centers are treated and compared to queueing approaches in [5] and further developed in [43]. Abandonments and retrials are also considered in [55] in a fluid

setting. The fluid models do not work well in underload situations however, as the stochastic element is disregarded. Fluid models are used for a multi-server setting with abandonments in [75]. Staffing of time-varying queues to achieve time-stable performance is treated in [24]. A fluid approximation that takes balking, impatience, and retrials into account is presented in [2].

In [24] a simulation based approach is used to determine staffing levels for inhomogeneous Poisson arrivals, general service times and general customer abandonment rates, referred to as $M_t/G/s_t + G$. The presented iterative-staffing algorithm (ISA) involves running a series of simulations based on the time-dependent arrival rate and iteratively adjusting the number of servers, s_t, according to a set target delay probability. The algorithm is shown to perform well for a realistic case.

Yet another approximation for systems with time varying arrival rates is given in [42], which uses a normal approximation and performs well for both slowly and quickly changing rates.

Another approach is to use an effective arrival rate at certain time instants determined by exponentially smoothing an estimated arrival rate. In this way steps in the arrival rate function can be taken into account, which seem to yield better results than the point wise stationary approximation. This is due to the fact that it can take the inertia of the system into account, i.e. after a step down in the arrival rate function, performance will be worse than the current rate would otherwise suggest and vice versa for steps up. Abandonments, retrials with some probability, and non-exponential service times (by using Erlang distributions) are also treated in [68]. A similar system is treated in [55].

Scheduling

Having determined the number of required agents for each time frame (e.g. half hour period), the next step is scheduling, i.e. making work rosters for the individual agents. This will often be done taking the employees preferences into account. Some may prefer to come in early in the morning while others may prefer working at night. Also breaks, seminars, and work-hour regulations need to be worked into the work schedule.

Scheduling is a well established and described discipline within operations research and available in many commercial products. The mathematical methods used for this purpose include simulated annealing, genetic algorithms, and other metaheuristics. The problems are often not solved to optimality due to their complexity but suboptimal solutions are seen as adequate. Scheduling including breaks and reliefs is e.g. treated in [64] where schedules are first made

without breaks, which are then included and additional agents added as necessary. Another approach based on simulation and cutting plane methods is presented in [7] and [19].

Sometimes it may not be possible to make a schedule that matches the required workforce for all time periods. However if short periods (e.g. 15 minutes) are used for scheduling it may be acceptable to have a slight understaffing in a few periods, which may be made up for by overstaffing in other periods. To fully account for this, staffing and scheduling need to be considered as one unified problem. This makes for very complex problems which may be difficult to solve for large systems. A fluid flow based model for staffing presented in [35] should be able to take possible schedules into account. Other work dealing with combined staffing and scheduling is discussed in [25].

4.3 Multi-Skill Call Centers

With the advent of more advanced ACDs, the use of skills-based routing is now the norm in practically all but the smallest call centers. Agents are divided into different groups according to their skills, e.g. the languages they speak, their IT-competence or financial knowledge. Incoming calls have to be routed based on some criteria such as the dialed number (by DNIS) or the number dialed from (by ANI). Another or additional way to determine how a call should be routed is from caller input into the IVR such as the customer entering his bank account number or choosing a desired service.

The reasons for implementing a multi-skill/class environment in a call center are many. An obvious reason is a desire to differentiate customers and give them different priorities. Examples of this are that some customers may be more valuable than others to e.g. a bank and some agents may have better competencies to handle the important customers. Another example is support lines that have two numbers customers can call, one free of charge and another charged number, where customers willing to pay are put ahead in line.

An additional and perhaps even more important reason for having multi-skill call centers is that it allows for a higher level of specialization of the agents. Specialized people generally perform better within their speciality [66]. However, dividing agents into smaller skill groups obviously negates some of the benefit gained by having a larger more effective unit. Allowing customers to be routed between the different skill groups based on a given policy can to some extent alleviate this dilemma by introducing some flexibility.

The use of skills-based routing complicates most of the aspects of call center management. As seen in Figure 4.2 the structural design of call centers can become very complicated indeed. Staffing becomes much more difficult as the number of agents assigned to one agent pool may influence the number of agents needed in another pool. Staffing of multi-skill call centers has experienced a growing interest in the literature in recent years. This topic is dealt with in e.g. [71], [35], and [62], while a recent overview can be found in [51].

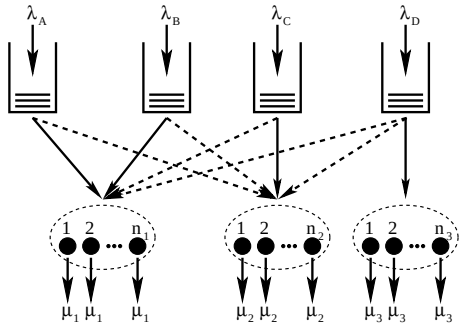


Figure 4.2: Multi-skill call center. Solid lines represent default call routing, dotted lines alternative routes.

The general consensus is that simulation is the only feasible way to analyze multi-skill call centers when they start to get just a little complicated, see e.g. [8]. The alternative approach to analyze more advanced routing schemes has been to split them up into smaller parts and introduce a number of canonical designs which are easier to deal with [25]. A number of canonical designs are shown in Figure 4.3.

While the canonical structures are often used to gain insight into parts of a more complicated call center setup, it still remains to be investigated how large the effect of these simplifications is. The effect will obviously depend on the individual setup, but caution should indeed be exercised when using these simplifications. This would be an interesting topic for a simulation study. It will depend on how large the influence of the ignored parts of the system is. If higher priority calls routed to a considered agent pool are ignored, the effect may very well be much larger than if lower priority calls are ignored. This is treated further in Section 6.3.

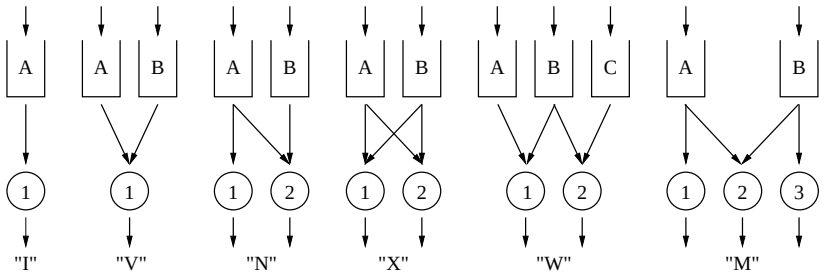


Figure 4.3: Canonical designs for skill-based routing as defined in [25].

The benefit of having cross-trained agents, such as in the "M"-design, has been thoroughly studied in the literature. The general consensus is that a few cross trained agents can make a system perform almost as well as a system with all cross trained agents [48]. This is a way to reduce costs as agents with more skills can be assumed to be more expensive than those with less.

The "M"-design is also investigated in [67], where one customer class is given non-preemptive priority over the other, when assigning calls to the generalists. Balking and abandonments are taken into account and simple performance measures calculated. A 4-dimensional state space is defined and the balance equations are solved using the power method. This approach is limited to smaller systems due to the curse of dimensionality problem.

In [59] it is shown that the Erlang-B formula is also valid for hyper-exponential service time durations in loss systems, which may occur if calls originate from different types of customers. Indeed, the Erlang-B formula is valid for any service time distribution, as only the mean of the service duration affects the blocking probability [37]. Hyper-exponential service time durations in delay systems are treated in [59].

A thorough treatment of multi-skill call centers is given in [61]. Limited computer resources in a multi-skill environment are treated in [3]. A fluid approximation to multi-server schemes with abandonments is presented in [75]. As always the fluid model works best for heavily loaded systems and it is concluded that the service time distribution mostly affect performance through its mean, whereas the abandonment (or patience) distribution affects performance through higher order moments to a higher degree. In [33] staffing of multi-class call centers (one agent skill, multiple customer classes) is addressed and the findings indicate that it may be enough to consider the total rate of incoming

calls.

4.4 Skills-Based Routing

In call centers with multiple customer classes and different agent skills, a central problem is how to route calls. This is often divided into call and agent selection policies. Call selection being how agents select the next call when they have finished serving a call, and agent selection being how incoming calls are routed if multiple agents are free.

Skills-based routing is obviously strongly intertwined with the other aspects of call center planning. However, the discipline of agent/call selection under a given workforce and call load poses many interesting issues. Many different routing schemes are used in call centers, as illustrated in Figure 4.3, but the consequence of using these are not fully understood.

The call selection and agent selection problems are treated in [6] for a skills-based setting. A system with one server pool and multiple customer classes is considered, where costs can be associated with different quantities such as queue length, abandonment and number of customers in the system. In most cases it is desirable to use a work conserving policy, however in certain cases it may indeed be desirable to have idle agents even while calls are waiting. In Appendix C it is shown that especially for the case where the cost functions are step shaped, e.g. zero below a given value and non-zero above it may indeed be advantageous to keep some servers free.

Agent Selection

Agent selection can further be divided into which agent pool calls are routed to and which individual agent in the pool gets the call. The most common practice in call centers is to route incoming calls to the agent who has waited the longest within an agent pool. This can be seen as the most fair policy for the agents. However, in a contact center where agents have other tasks than answering calls it may be better to use a different scheme in order to give agents fewer but longer breaks. This would allow them better to be able to concentrate on other tasks (answering emails, etc.) and is certainly an interesting topic for further research.

Agent pool selection becomes relevant when calls are allowed to go to more than one agent pool and agents are free in these. This could be incoming A-calls in the "N"-design, but not B-calls. Or B-calls, but not A- and C-calls

in the "W"-design. The standard way to implement agent pool selection is to have a preferred agent pool for each call type, this could be pool number 1 for A-calls in the "N"-design. Often it would be preferable to send calls to agents who are most specialized in order to let more generalized agents be available to handle other call types. This means that A-calls should be sent to agent pool 1 and B-calls to agent pool 3 if possible in the "M"-design. Agents in pool 2 should thus only be assigned calls if none of the respective specialized agents are available. An approach for optimizing routing policies in complex setups based on one-step policy improvement is presented in [12]. No queues are allowed in the system thus the objective is to minimize blocked calls which is effectively done.

Another implementation used in call centers is only to allow calls to go to a server group after a certain fixed time. The motivation for this type of setup could be if a call center has a front and back office, where it is preferred that front office agents take the calls. Back office agents may have other tasks and only be used when a queue builds up in order to improve the service levels. This type of policy is treated further in Chapter 5.

A multi-skill system without waiting room is treated in [52] in order to examine the loss probability and overflow process. They treat setups with many skills and customer types and come up with an approximation that is used to analyze the performance of different routing policies, i.e. the order of which incoming calls select an agent type.

Call Selection

Call selection becomes relevant when an agent finishes serving a customer and is allowed to take multiple classes of calls, i.e. agents in the "V"-design or agent pool 2 in the "N"-design. Within each of the queues, the norm is to answer calls according to FCFS for fairness, even if other policies may improve service levels as discussed in Section 2.3.

A straightforward approach to call selection is to rank customers strictly according to different priorities. Early work on this policy is presented in [22]. This is also referred to as head-of-line prioritization, as high priority customers are allowed to get in front of lower priority customers. When an agent finishes a call, the next call routed to the agent will be the one waiting with the highest priority. This policy may, however, lead to very long waiting times for lower priority customers, especially if the majority of customers are of high priority.

Using dynamic priority can to some extent overcome the risk of exceedingly long waiting times for low priority customers. This prioritization scheme is

implemented by adding a constant, k_X, dependent on the priority class, X, to the waiting time of customers. When an agent becomes free, the next call he will receive, will be the one with the lowest combined value of the waiting time and the constant. The added constant can in this way be seen as a target waiting time, and lower values of the constants correspond to higher priority. This policy was first described in [38] in its discrete form and is sometimes also referred to as scheduling according to due date. The tail behavior is described in [39] as being exponential, and the first part of the distribution is examined using The Erlang Approximation (TEA) in Appendix B.

Service according to the dynamic priority discipline is bounded by FCFS and head-of-line prioritization. If different classes have the same priority constant they will obviously be served according to FCFS. At the other extreme, if a customer class' priority constant is allowed to tend to infinity, then it will always get service before other classes, which would correspond to strict or head-of-line prioritization.

The dynamic priority discipline is referred to as being endogenous [40], because the order in which customers are served not only depends on their priority class but also on the waiting time the individual customer has already incurred. A prioritization scheme based on a somewhat similar principle is introduced in [46] and further treated in [47]. Instead of adding/subtracting constants as in dynamic priority, the waiting time is multiplied with different factors based on relative priority, and the customer with the highest product is chosen for service.

The fixed threshold policy discussed for agent selection also applies for call selection policies. It implies that agents are only allowed to take calls from certain queues that have waited at least the given threshold. This is typically combined with different priorities in the normal sense. An example could be to make agent pool 2 in the "N"-design of Figure 4.3 take B-calls, unless there is a waiting A-call that has waited more than the given threshold. This is a common policy in call centers, probably due to ease of implementation. Also it is convenient to relate the fixed thresholds to TSF, however this relation is not really backed by anything but assumptions. Indeed it is not straightforward, and it is certainly not documented in the literature until now. It is treated in Chapter 5 and Appendix C.

CHAPTER 5

Waiting Time Based Routing Policies

Routing policies based on a fixed or deterministic threshold on the waiting time of the first customer in line are used widely in real call centers but the effects of using these are not very well understood or described in the literature. The three papers in appendices A, B, and C examine routing policies with waiting time based policies, in particular, fixed thresholds.

The reasons for using fixed thresholds may be many. One is prioritization of calls by letting some calls wait longer than others, before overflow is allowed. The most obvious reason is probably in relation to service levels in the form of TSF. In this way the fixed threshold can be used to send a call to another agent group, just before its waiting time reaches the SLT, in order to keep service levels high. The reasoning behind this is that a call has a certain amount of time to be routed to the agent group for which it was intended, while still keeping the possibility of letting it overflow to avoid excessively long waiting times. The actual effects of using the policy are analyzed in Appendix C.

Fixed type threshold policies have been treated in the literature, however the threshold has been based on the number of customers in queue rather than on the waiting time. Such a setup is analyzed in [36]. There is an evident advantage to base the threshold on the waiting time of the first in line as compared to the queue length. A queue-length based threshold is strongly

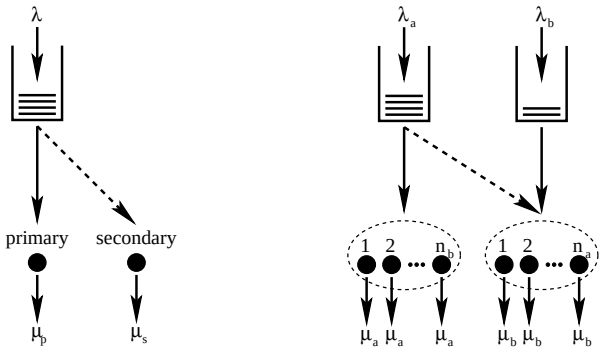
dependent on the service times and number of servers. Take as an example a threshold that allows overflow when 20 customers are waiting, this may not be a strict threshold during peak hours where maybe 100 agents are working, whereas during the night where traffic is low and maybe only a few agents are on duty, the overflow threshold would be very high in terms of waiting time.

The lack of literature on fixed threshold policies based on the waiting time can be ascribed to the difficulties of dealing with constant values in stochastic systems. Most variables in models are normally assumed to follow the exponential distribution or other closely related distributions, because these are easier to deal with due to the lack of memory property. As the policy is widely used in call centers, it is essential to obtain a better comprehension of what the implications of using it are.

The natural and most used approach to model queueing systems, since the times of Erlang, has been to base the analysis on the number of customers in line. Indeed a fixed threshold overflow policy is analyzed in [9] based on this approach. Here, states of the Markov chain correspond to the number of customers in queue, and state dependent overflow probabilities are used to emulate a fixed threshold. The fixed threshold in this case determines whether an incoming call is allowed to go to a back-office group of agents or not. The decision to allow overflow or not is thus taken immediately upon arrival from a straightforward presumption that the risk of having to wait more than the given threshold increases with the number of customers already in queue. The approach seems to work well and an approximation of the waiting time distribution together with other performance measures are obtained. Apart from [9], the literature on fixed thresholds on the waiting time is non-existent.

The three papers presented in appendices A, B, and C deal with fixed thresholds on the waiting time and are all based on a novel approach. Instead of considering the number of customers queued, the waiting time of the first customer in line is used as the primary variable. This approach allows fixed thresholds to be dealt with directly, contrary to the usual approach. In all cases the waiting time distributions are derived or approximated. Obtaining the waiting time distributions is essential to fully understand the dynamics of the system and to relate the results to the use of TSF as a performance measure in call centers.

The simple design of Figure 5.1a is analyzed analytically in Appendix A and discussed in Section 5.1. An approximation capable of dealing with the more complex setup in Figure 5.1b is presented in Appendix B and discussed in Section 5.2. In Appendix C overflow in systems with "N"-design is optimized, this is also discussed in Section 5.3.



(a) "IV"-design **(b)** "N"-design

Figure 5.1: The setups with fixed thresholds, which the work presented in appendices A-C is centered around. The simpler setup in (a) is analyzed analytically whereas approximations are used for the more involved multi-server setup in (b).

5.1 An Analytical Approach

The simplest possible setup using the fixed threshold on the waiting time of the first customer in line (W_t) is analyzed in Appendix A. An illustration of the setup is shown in Figure 5.1a. Only having one primary server (front office agent) and one secondary server (back office agent) allows an analytical solution of the waiting time distribution to be found. Calls are only allowed to go to the back office (secondary) server when its waiting time has reached or surpassed a fixed value K. Both service times and arrivals are assumed Markovian.

Put briefly, the analysis is based on the evolution of the waiting time of the first customer in line. This First In Line (FIL) process is illustrated with an example in Figure 5.2. W_t obviously increases linearly with time, when at least one customer is queued. Whenever customer i enters service, the waiting time of the first customer in line decreases with an exponential value, A_i, as the time between arrivals is exponential. If W_t in this way decreases to a negative value, it is taken as the customer who just initiated service was the only one in the queue, and W_t is set to zero. This process is analyzed using forward Kolmogorov equations and level crossing arguments. In this way eight linear equations are produced. The solution to this equation system gives a set of constants, which are included in the final solution of the waiting time

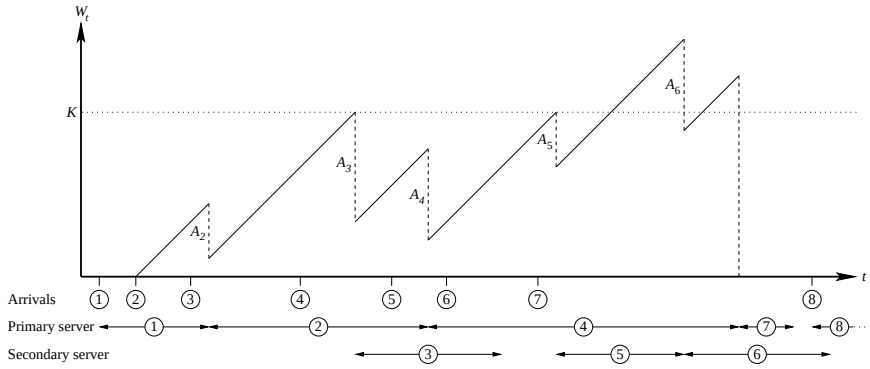


Figure 5.2: Illustration of the waiting time of the first customer in line process for the setup shown in Figure 5.1a. The numbers in circles illustrate arrivals and service times of customers.

distribution.

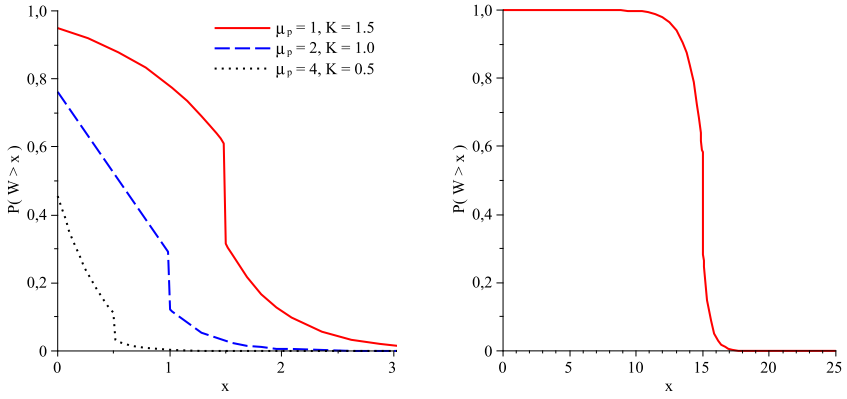
As mentioned, the solution is based on solving a set of eight equations. This equation system has been shown to be solvable for all conceivable parameter values, however general independence of all the equations has not been shown. What has been shown though, is that at least seven of the eight equations are independent, this is done in Appendix A.1.

The form of the resulting waiting time distribution is extremely simple and attractive as it is in a way intuitively evident. The density of the waiting time customers experience in the system, $w^{\mathrm{FIL}}(x)$, is given in Equation 5.1. The constants c_1 and c_2 are determined from solving the equation system discussed earlier.

$$w^{FIL}(x) = \begin{cases} c_1 e^{(\lambda-\mu_p)x}, & \text{for } 0 < x \leq K; \\ c_2 e^{(\lambda-\mu_p-\mu_s)x}, & \text{for } x > K. \end{cases} \tag{5.1}$$

As is seen, the shape of the waiting time distribution takes the same fundamental form as for an $M/M/1$ queue, i.e. exponentially distributed waiting times for values below and above K respectively. An additional atom at the overflow threshold K corresponds to the customers allowed to go to the secondary server because their waiting time reached K.

For the front and back office setup there is an interesting consequence of the fact that higher traffic is allowed than what the front office server can handle. The consequence of this is that the waiting time distribution for values less than K becomes concave when $\lambda > \mu_p$. This is illustrated in Figure 5.3a. If the primary server is highly overloaded and the overflow threshold is large, most calls will experience a waiting time close to K as illustrated in Figure 5.3b. Obviously the traffic must be less than the combined capacity of the two servers in order for the system to be stable as abandonments are not considered.



(a) $\lambda = 2$, $\mu_s = 3$ **(b)** $\lambda = 2$, $\mu_p = 1$, $\mu_s = 3$, $K = 15$

Figure 5.3: Analytically determined waiting times for the two server setup shown in Figure 5.1a.

Even though the direct practical use of this result may be limited, it is theoretically very interesting, as it may indicate that a similar simple result would apply to a multi-server setup. This hypothesis does require further research though. Also the results can be seen as a proof of concept of the FIL approach.

5.2 The Erlang Approximation

The analytical approach used for the two server setup treated in Appendix A is not easily applicable to more complex setups, as the number of equations and the resulting expressions increase very quickly with the complexity of the

system. An example of this is that a 2-dimensional FIL process would need to be analyzed, if a system with two queues was to be considered. In Appendix B The Erlang Approximation (TEA), based on the same fundamental idea of modelling the waiting time of the customer first in line, is presented.

A discrete approximation of the FIL process is used, instead of modelling it explicitly as in Appendix A. This allows the system to be modelled as a Continuous Time Markov Chain (CTMC) through which waiting time distributions can be obtained. Besides modelling the waiting time of the first in line, the number of free servers is also included in the model thus allowing multi-server setups to be treated.

The principle behind TEA is illustrated in Figure 5.4. Whenever servers are free the system is modelled in the usual way of keeping track of the number of free servers, represented by negative indices. When customers start queueing up the FIL waiting time is modelled by states with positive indices. Transitions with rate γ represent evolving time. The number of steps backward, when a customer goes into service, follows a truncated geometric distribution illustrated by the double arrows in Figure 5.4. These correspond to the exponentially distributed jumps in the continuous representation described in Section 5.1. A truncated state space is used for the actual computations.

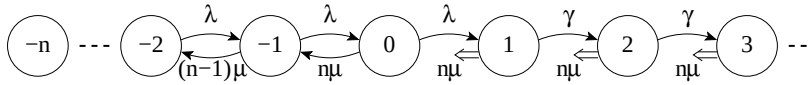


Figure 5.4: The Erlang Approximation illustrated by states and transitions of the Markov chain for the relatively simple $M/M/n$ system.

The waiting time of customers is approximated by considering the states from which customers enter service. For example, if a customer enters service from state i, this corresponds to having waited a sum of i exponentially distributed (with rate γ) time frames, i.e. a gamma distributed amount of time. The approach used to determine the waiting time distributions is described in detail in Appendix B.

When TEA is used for system with "N"-design, the underlying Markov chain becomes 2-dimensional. The combination of the number of free servers (negative indices) and the FIL waiting time (positive indices) in each dimension helps reduce the curse of dimensionality problem and makes the approximation of the "N"-design feasible. Systems with three or more server groups and

customer classes could in principle also be analyzed using TEA, but then the curse of dimensionality would quickly become a problem.

TEA is shown to converge to the analytically correct solution when used for an $M/M/1$ system in Appendix B.1. It converges for $\gamma \to \infty$ and when the number of states with positive indices representing the FIL waiting time is infinite. This result nicely supports the fundamental approach of TEA. Simulations are used to verify TEA for the more complex setups.

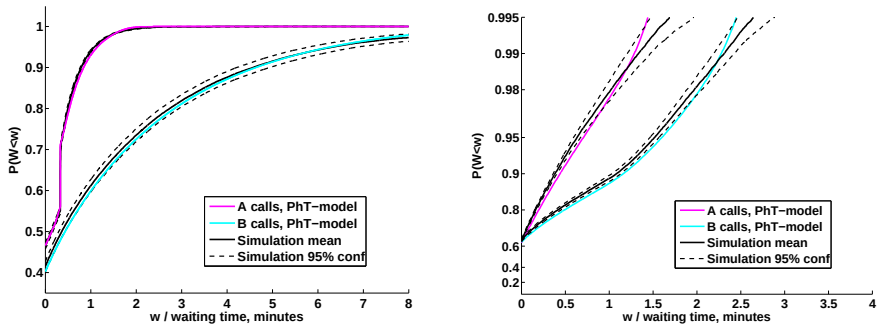
TEA is first and foremost used to model the "N"-design shown in Figure 4.3 and obtain the waiting time distribution from which relevant performance measures such as ASA and TSF are easily calculated. An example of the waiting times as found through TEA is presented in Figure 5.5a and compared to simulations.

The generic nature of TEA is also demonstrated in Appendix B by analyzing other models such as systems featuring dynamic priority, which were discussed in Section 4.4. A conjecture in [39] states that the tails of the waiting time distributions have an exponential shape for a system based on the "V"-design with dynamic priority. This is based on the fact that if customers arrive at a system where they can expect to wait a long time, then a B-customer that has already waited $(k_A - k_B)$ will receive the same service as a recently arrived A-customer. The conjecture is illustrated in Figure 5.5b where a logarithmic plot is used to illustrate the exponential tails. TEA is shown to work best for the lower and most interesting part of the waiting time distribution. The deviation of the model from the simulations in the upper part is magnified by the logarithmic plot.

5.3 Optimization of Overflows

The next obvious question is whether the fixed threshold policy is actually attractive and if it is, for which scenarios. This is treated in Appendix C, where the "N"-design from Appendix B is further analyzed. The approach used for the optimization is based on adding decisions to the states in the underlying Markov chain of TEA. In this way it can be treated as a Markov Decision Process.

The decisions subject to optimization are in regard to how agents in server group 2 of the "N"-design choose calls. When an agent in this group becomes free and calls are waiting in both queues it must choose whether to take an A- or B-customer into service. Likewise, if only A-calls are waiting it must choose whether to serve this or leave it. The reason for not taking an A-call, while no



(a) "N"-design, fixed overflow threshold. **(b)** "V"-design, dynamic priority.

Figure 5.5: Waiting time distributions found from The Erlang Approximation.

B-calls are waiting, would be to keep the group 2 server(s) reserved for future B-calls.

In order to optimize the control of overflowing calls, costs must be assigned to certain events. In Appendix C this is done by imposing a cost whenever a call is taken into service after having waited in queue. The actual form of these cost functions depends on what service measures are prioritized such as ASA and/or TSF. Costs can also be assigned to whenever an A-call is taken by a group 2 agent, i.e. when an overflow happens. Depending on how the cost functions are given, both A- and B-calls can be of higher priority than the other.

It is shown that the fixed threshold policy is in some cases nearly optimal. This is the case if ASA is considered and A calls are given higher priority than B calls and an additional cost is assigned to overflowing calls.

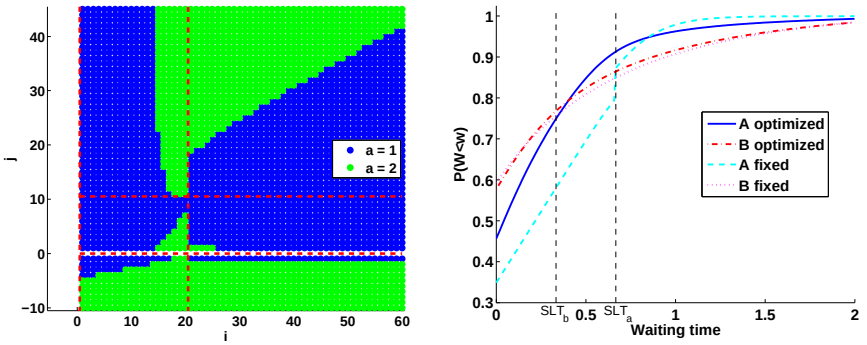
The optimal policies when only considering ASA take some fairly simple forms. In general there should almost never be free servers, if calls are waiting. If calls from both classes are waiting and an agent from pool 2 becomes free, he should generally take the customer having waited the longest. Different parameters may affect the optimal policy in favour of either customer class though.

When TSF is used as basis for the cost functions, the picture becomes much more involved. The optimal policies become rather complex, however, they can be well explained by closer inspection as done in Appendix C.

An example of the optimal policy when the optimization is based on TSF

is seen in Figure 5.6a. Each marker corresponds to a state in the state space of the underlying Markov chain of TEA. The x-axis represent free servers in agent group 1 (negative indices) and the FIL waiting time of queue A (positive indices), the y-axis represent server group 2 and the B-queue in the same way. Green markers mean that A-calls should be sent to server group 2, blue markers they should not. The white areas show states where no decision is possible. As is seen no decisions are possible when group 1 servers are vacant. Whenever A-calls are queued, decisions are tied to each state, except when all group 2 servers are occupied and no B-calls are waiting. This is due to a technical modelling issue explained in the detailed presentation of the optimization approach in Appendix C.

Figure 5.6b shows the resulting waiting time distribution of the optimized policy compared to having a fixed threshold policy. It is seen that the tail of the optimized policy is larger than that of the fixed threshold policy for A-calls, but more calls are served within the SLT (the rightmost vertical dotted black line) when using the optimized policy. As the optimization is only based on the fraction of calls served within the SLT, this is as desired.



(a) Optimal policy. **(b)** Waiting time distributions.

Figure 5.6: The resulting policy from optimizing with regard to TSF (green means send A-call to server group 2, blue means do not) and the appertaining waiting time distributions comparing fixed and optimized thresholds.

All in all, Appendix C shows that the fixed threshold policy may indeed be attractive to use, however it can be improved by more complex policies in most cases.

CHAPTER 6

Simulation Modelling

The very complex nature of modern call centers makes it difficult if not impossible to construct analytical models that describe their operation and performance in detail. Discrete event simulation is often mentioned as the only feasible approach, when the performance of multi-skill call centers needs to be investigated [57], [25].

Modelling a call center by discrete event simulation enables very complex setups to be analyzed. This includes call centers with multi-skill setups, general service time distributions, time varying arrival rates, and a varying number of agents working during the day.

A detailed simulation model can in many cases represent a real scenario more closely than analytical models, especially for complex setups. A simulation model built on a real scenario is thus suitable to analyze what happens if a certain parameter, such as the threshold determining when overflow is allowed, is tweaked. This may be the case even if the simulations are not capable of reproducing the real performance data precisely, a general trend may still be easily identifiable.

One of the drawbacks of simulations is that it can be hard to restrict the extent of the investigation to something that produces transparent and useful results due to the many parameters involved in complex models. In order to structure the investigations, proper "design of experiments"-techniques should be used, see [58] and [45] for exhaustive texts on this topic.

Another drawback is that running a sufficient number of simulations to obtain reliable results can be time consuming even on high performance computers. Also the implementation of a complex model can require a lot of work.

There is an abundant amount of literature on discrete event simulation both in the general setting and within the call center world. The role of simulation as a tool to aid in decision making within complex multi-skill call centers is discussed in [8]. The fundamental issues connected to call center simulation such as relevant inputs and outputs are described [57] and [56]. Simple examples of the possibilities of simulation modelling that deal with cross training of agents in multi-skill call centers are also given in both of these references.

A real-world scenario involving an outbound call center is analyzed using simulations in [32]. Three different call scheduling methods, including the one actually used in the call center, are compared and it is shown that one of the alternative methods will result in more Right Party Contacts (RPCs).

Within this chapter the basic elements of discrete event simulation are described in Section 6.1. The different approaches that can be taken regarding the use of software for simulation are discussed in Section 6.2. Finally in Section 6.3, an example is given of how simulations can be used to investigate the effect of relying upon canonical model structures.

6.1 Discrete Event Simulation Basics

A realistic discrete event simulation model of a call center involves a lot of parameters. Some of these are normally regarded as uncontrollable factors, such as call patterns and service times. Obviously measures can be taken to modify these, but basically this involves much more than changing a parameter somewhere in the telephony system. On the other hand, controllable factors include routing policies, agent schedules, and the assigned agent skills.

Based on these parameters, a discrete event simulation model simulates the operation of a call center down to the individual calls arriving, perhaps waiting, and finally being served by an agent or abandoning. An illustration of how the handling of calls by individual agents proceed in a simulation of a multi-skill call center is shown in Figure 6.1. The illustrated simulation is based on a setting similar to the one in Figure 4.2, i.e. a setting with four different call types, three agent groups, and overflow between these. Figure 6.1 also illustrates how a simulation model easily can be adapted to account for different arrival and service rates, different number of agents in the groups, and specific routing policies.

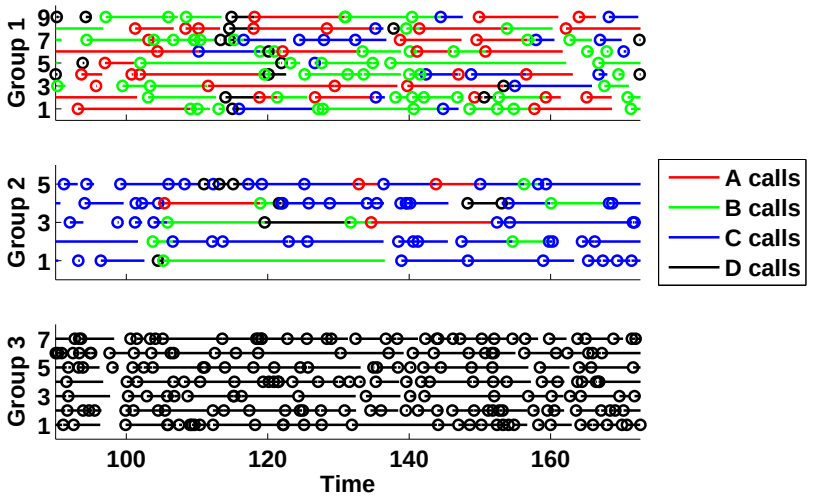


Figure 6.1: Illustration of the details of agents' work in a discrete event simulation of a multi-skill call center as the one seen in Figure 4.2. Each line corresponds to an individual agent, the circles show when agents pick up calls, and the lines to the right of the circles show the service times.

The term discrete event simulation refers to how the software program takes all events into account. These events could include an arriving call, a call having waited a certain time that allows it to overflow, and agents finishing serving a call. The events are then typically simulated in the order they happen, and the times at which they happen are updated and recorded for analysis.

Service and inter-arrival times are typically assumed to happen according to a given distribution from which the individual values for the calls are sampled. This sampling of values relies heavily on random number generation, thus simulations of the same system will give different results if not the same seed is used. This means that the more simulations are carried out, the more precise the result can be assumed to be and indeed confidence intervals can and should be determined. In this way simulations can be used to verify the correctness of analytical results as is done in the work presented in Appendix A. Simulations are also often used to examine how good approximations are, examples of this are seen in Appendix B and in [52].

Relevant data should be collected during the simulations in order to be

able to assess the performance of the simulated system. Figure 6.2 shows the number of occupied agents and the number of different calls in queue for a simulation of a multi-skill system as an illustration of this.

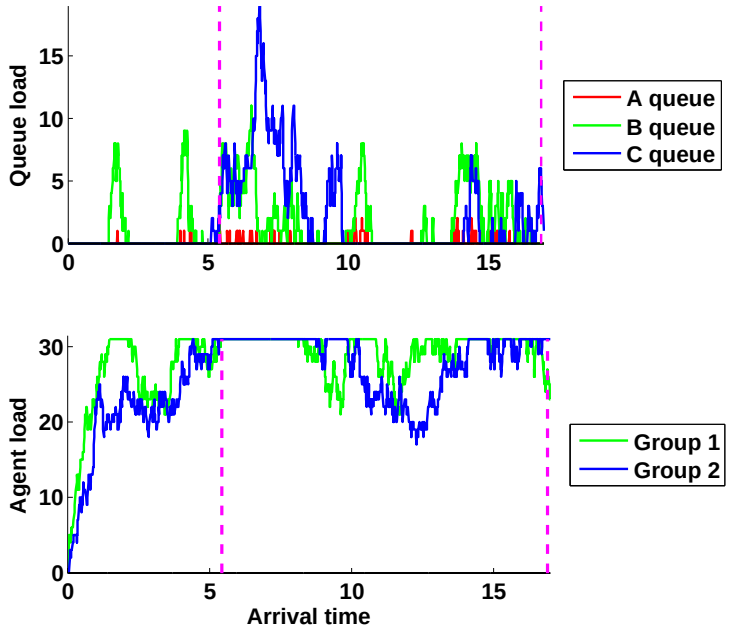


Figure 6.2: Illustration of the load on a simulated system and the transient that needs to be taken into account when starting with an empty system. Only events within the vertical dotted lines should be used to calculate performance measures.

What is seen from Figure 6.2 is that it takes some time for the system to reach a steady phase. An often used approach is to start with an empty system and disregard the transient phase until the system reaches a stable phase. How long this transient phase should be, is ambiguous, however, as a starting point it is better to make it too long than too short. A somewhat similar modification of the simulated data must be carried out at the end of the simulation. Here it is appropriate to remove all events after the last arrival, as low priority calls queued after this time instant will never be overtaken by high priority calls and thus in principle receive better service.

The collected data forms the basis for the outputs of the simulation such

as TSF, ASA, and fraction of abandoned calls. These outputs can be used to examine what effect different changes of the controllable factors will trigger.

6.2 Choice of Software

Simulation of call center performance can be approached in different ways. A fundamental decision is whether to build a simulation model from scratch or use a commercial software package. Building a simulation model from scratch using a programming language such as C, C++, Java, Matlab, etc. requires a certain level of familiarity with stochastic processes and discrete event simulation principles together with the obvious programming skill. Contrary to this, using a commercial software package such as Arena, Extendsim or OPNET should be much more approachable, as these will typically provide a graphical user interface. It should also be taken into account that acquiring the license for a commercial software package can be quite costly.

In [70] simulation of a call center using C and Arena is compared. In general the C-implementation is favoured by the authors due to its faster performance and better ability to deal with the finer details in a call center setup. An example of the difficulties of dealing with details in the commercial software is the possibility to assign calls to the free agent who has waited the longest, which proved troublesome in Arena but straightforward in C. However, for most call center managers' point of view, Arena, or other dedicated commercial simulation packages, may indeed be preferable due to the much faster and easier model building process. In this perspective it should also be pointed out that managers in commercial call centers may not have the time and perhaps not the skill to develop a simulation program from scratch.

The aforementioned possible lack of programming skills among call center managers is also addressed in [63] where an implementation of a simulation tool with an Excel interface is presented. The underlying simulation engine is based on a combination of Visual Basic and Arena. Despite the obvious advantage of a familiar interface, the model only encompasses a single-skill call center. Building a flexible multi-skill simulation tool in this way quickly becomes much more involved than what is presented in [63]. Using such an approach will also still require a costly license for the simulation software.

An in-between approach, between the full-blown commercial software packages and from-the-scratch implementations, might be to have a dedicated simulation model built, based on a given call center setup. This could eventually have an Excel/VB interface as above combined with a dedicated program writ-

ten in e.g. C. For a multi-skill call center the most feasible solution using this approach would probably be to have a dedicated model where routing policies are somewhat fixed, but the individual variables, such as call load, number of agents, etc. can be tweaked as desired.

6.3 Illustrative Simulation Example

A small example of how discrete event simulation can be used to investigate a concrete issue is presented in this section. It is in no way an exhaustive analysis, but rather just a teaser of the possibilities. The simulation model was implemented in Matlab.

Consequence of relying on canonical designs

In Section 4.3, it was discussed how one should be careful if relying too much on performance measures obtained from analyzing the simplified canonical structures shown in Figure 4.3. In order to investigate the consequence of ignoring a part of a multi-skill setup, the system in Figure 6.3 is analyzed using a discrete event simulation model.

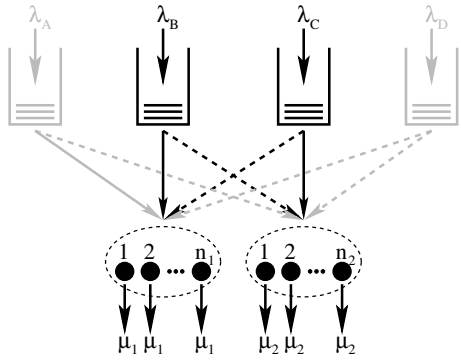


Figure 6.3: The simulated setup, dotted lines show where the FIL waiting time must exceed a threshold, before calls are allowed to be taken from the queue.

The reference case corresponding to a simplified canonical "X"-design is shown in black in Figure 6.3. It includes two call types, B- and C-calls, and two server groups, group 1 and 2. This reference case is compared to a case

where lower priority calls are added to the system, referred to as D-calls and illustrated with grey in the right part of Figure 6.3. The last case examined is one where high priority calls are added, corresponding to the A-calls in the figure, also shown in grey. In this case no D-calls are present.

The parameters used for the simulation are given in Table 6.1. Arrival rates, service times and abandonments are all assumed Markovian. The units can be arbitrary as long as they are consistent. In the call center setting they could very well be minutes (and 1/minute). The arrival rate of calls of type i is given as λ_i, service rates of agents in group j as μ_j, number of agents in group j as n_j, abandonment rate of type i calls as α_i and finally the fixed thresholds on the waiting time of the first in line before overflow is allowed for call type i is given as k_i.

Table 6.1: Parameters for the simulation.

Parameter	λ_A	λ_B	λ_C	λ_D
Reference case	0	27	27	0
Added low priority case	0	27	27	6
Added high priority case	6	27	27	0
Shared parameters	$n_1 = 30$, $n_2 = 30$, $\mu_1 = 1$, $\mu_2 = 1$, $\alpha_{A,\ldots,D} = \frac{1}{3}$, $k_{A,\ldots,D} = \frac{1}{5}$.			

As is seen, the three cases only differ in that a small amount of D-calls are added in the second case and a small amount of A-calls in the third case. In this way, the total traffic offered to the constant number of servers is not the same in the reference case as in the other two cases. As the intent is to investigate the consequences of ignoring a part of the setup altogether, this is as it should be.

In all cases 200 iterations of 10′000 calls were simulated and the average values of the performance measures calculated. This amount of simulations gives only a small uncertainty on the last digit of the presented results. In this way, the uncertainties are nowhere near significant as compared to the differences between the service measures of the three cases.

The first output of the simulations is the percentage of calls served within 1/3 time unit corresponding to 20s if minutes are used as time unit. The percentage is calculated by taking the number of calls served within the time limit

divided by the number of offered calls, thus abandoned calls are also counted in the denominator as discussed in Section 2.3. The other two performance indicators shown, are ASA and the percentage of abandoned calls. The average values of the results from the 200 simulations are shown in Table 6.2.

Table 6.2: Simulation results. Input parameters are assumed to be in minutes.

Call type	A	B	C	D
Service levels with 20s SLT.				
Reference case	-	98.7%	95.5%	-
Added low priority case	-	97.6%	88.9%	54.1%
Added high priority case	99.0%	95.9%	75.9%	-
Average Speed of Answer				
Reference case	-	2.27s	3.15s	-
Added low priority case	-	4.17s	6.64s	30.7s
Added high priority case	1.84s	6.36s	10.4s	-
Lost calls due to abandonments				
Reference case	-	1.28%	1.76%	-
Added low priority case	-	2.32%	3.73%	17.8%
Added high priority case	1.04%	3.52%	5.99%	-

It is seen from Table 6.2 that adding high priority calls affects the three shown performance parameters of B- and C-calls to a much larger degree than adding low priority calls. Obviously, this can also be interpreted as ignoring high priority calls, when relying on canonical structures induces a much larger error than ignoring low priority calls.

The results presented in this section correspond nicely to what could be expected. They illustrate in a tangible way that caution and common sense should be exercised if relying on a simplified model of a larger more complex system. Furthermore the results give a small example of the possibilities of using discrete event simulation for modelling call center performance.

CHAPTER 7

Discussion

The content and contributions of this thesis are summarized and discussed in this chapter.

In chapters 2-6 of the thesis, the fundamental methods and issues of call center management are treated. This spans from a general description of call centers and their structure over data analysis to capacity planning using queueing models and simulations. A thorough literature review together with numerous references are given throughout these chapters.

The main part of the thesis is the three papers presented in appendices A, B, and C. The background of this work and these papers is discussed in Chapter 5.

The single main contribution is the development and use of the First In Line (FIL) analysis. This approach involves modelling the waiting time of the customer first in line as the main variable. This is a fundamentally new approach to modelling queueing systems and opens up new possibilities regarding analysis of certain queueing systems as discussed below.

The concept of FIL analysis is proven by the analytical approach taken in the paper presented in Appendix A. Two different uses of the approach are presented in this paper, one dealing with a single server that adapts its service according to the waiting time of the first in line. This analysis is interesting, because one could very well imagine a server that hurries up the current service, if a waiting customer becomes impatient after having waited a long time. The

other use treats a two-server setup where the secondary server is only allowed to take jobs that have waited more than a given fixed time. The FIL approach enables exact analysis of this system, which was otherwise problematic due to the inherent difficulties of dealing with deterministic entities in stochastic systems. As a result, a very simple and attractive expression is found for the waiting time distribution of customers in the latter system.

An approximation based on the FIL principle is developed in the paper presented in Appendix B and referred to as The Erlang Approximation. This approximation enables analysis of more complex systems than what is feasible using the exact approach of Appendix A. The versatility of TEA is demonstrated by approximations of the performance of different queueing systems, all where the waiting time of customers somehow plays an important role. Numerous suggestions for further uses are also discussed.

The last paper, presented in Appendix C, takes the use of the approaches developed in the two former papers a step further. Here, the overflow between server groups is optimized with regard to the service level measures used in the call center industry. Furthermore the appropriateness of the often used fixed threshold policy is verified and the implications of using it are discussed.

The work presented spans a nice range of research fields. From the very analytical approach in Appendix A, over the more application oriented approximation of Appendix B, to optimization of concrete call center processes in Appendix C.

The FIL approach should be taken into consideration whenever queueing systems, in which processes somehow depend on the waiting time of customers, are to be analyzed.

Bibliography

[1] The call center's incredible shrinking day. *Credit Union Executive Newsletter*, 28(29):2, 2002.

[2] Salah Aguir, Fikri Karaesmen, O. Zeynep Akşin, and Fabrice Chauvet. The impact of retrials on call center performance. *OR Spectrum*, 26(3):353–376, 2004.

[3] O. Zeynep Akşin and Patrick T. Harker. Modeling a phone center: Analysis of a multichannel, multiresource processor shared loss system. *Management Science*, 47(2):324–336, 2001.

[4] Z. Aksin, M. Armony, and V. Mehrotra. The modern call center: a multidisciplinary perspective on operations management research. *Production and Operations Management*, 16(6):665–688, 2007.

[5] Eitan Altman, Tania Jimenez, and Ger Koole. On the comparison of queueing systems with their fluid limits. *Probability in the Engineering and Informational Sciences*, 15(2):165, 2001.

[6] Rami Atar, Avi Mandelbaum, and Martin I. Reiman. Scheduling a multi class queue with many exponential servers: Asymptotic optimality in heavy traffic. *Annals of Applied Probability*, 14(3):1084–1134, 2004.

[7] Júlíus Atlason, Marina A. Epelman, and Shane G. Henderson. Call center staffing with simulation and cutting plane methods. *Annals of Operations Research*, 127(1-4):333–358, 2004.

[8] A.N. Avramidis and P. L'Ecuyer. Modeling and simulation of call centers. *Winter Simulation Conference, 2005 Proceedings of the*, page 9 pp., 2005.

[9] Wolfgang Barth, Michael Manitz, and Raik Stolletz. Analysis of two-level support systems with time-dependent overflow - a banking application. Forthcoming in *Production and Operations Management*, available from http://ideas.repec.org/p/han/dpaper/dp-399.html, 2009.

[10] Rene Bekker. *Queues with state-dependent rates*. PhD thesis, Eindhoven University of Technology, 2005.

[11] S. Bhulai and G. Koole. A queueing model for call blending in call centers. *IEEE Transactions on Automatic Control*, 48(8):1434–1438, 2003.

[12] Sandjai Bhulai. Dynamic routing policies for multiskill call centers. *Probability in the Engineering and Informational Sciences*, 23(1):101, 2009.

[13] V.A. Bolotin. Telephone circuit holding time distributions. *Fundamental Role of Teletraffic in the Evolution of Telecommunications Networks. Proceeding of the 14th International Teletraffic Congress - ITC 14*, 1:125–34 vol.1, 1994.

[14] S. Borst, A. Mandelbaum, and M.I. Reiman. Dimensioning large call centers. *Operations Research*, 52(1):17–34, 2004.

[15] Andreas Brandt and Manfred Brandt. On a two-queue priority system with impatience and its application to a call center. *Methodology And Computing In Applied Probability*, 1(2):191–210, 1999.

[16] L. Brown, N. Gans, A. Mandelbaum, A. Sakov, H. Shen, S. Zeltyn, and L. Zhao. Statistical analysis of a telephone call center: A queueing-science perspective. Technical report, University of Pennsylvania, 2002. downloadable at http://iew3.technion.ac.il/serveng/References/references.html.

[17] Lawrence Brown, Noah Gans, Avishai Mandelbaum, Anat Sakov, Haipeng Shen, Sergey Zeltyn, and Linda Zhao. Applications and case studies - statistical analysis of a telephone call center: A queueing-science perspective. *Journal of the American Statistical Association*, 100(469):36–50, 2005.

[18] U.S. Department of Labor Bureau of Labor Statistics. *Occupational Outlook Handbook, Customer Service Representatives*. 2008-09 edition. Available at http://www.bls.gov/oco/ocos280.htm.

[19] Mehmet Tolga Cezik and Pierre L'Ecuyer. Staffing multiskill call centers via linear programming and simulation. *Management Science*, 54(2):310–323, 2008.

[20] Brad Cleveland. Where is everybody?! how to accurately predict schedule shrinkage. *Customer Management Insight*, page 22, 2008.

[21] Brad Cleveland and Julia Mayben. *Call Center Management on Fast Forward: Succeeding in Today's Dynamic Inbound Environment*. Call Center Press, first edition, 1997.

[22] Alan Cobham. Priority assignment in waiting line problems. *Journal of the Operations Research Society of America*, 2(1):70–76, 1954.

[23] Richard A. Feinberg, Leigh Hokama, Rajesh Kadam, and IkSuk Kim. Operational determinants of caller satisfaction in the banking/financial services call center. *International Journal of Bank Marketing*, 20(4):174–180, 2002.

[24] Zohar Feldman, Avishai Mandelbaum, William A. Massey, and Ward Whitt. Staffing of time varying queues to achieve time stable performance. *Management Science*, 54(2):324–338, 2008.

[25] Noah Gans, Ger Koole, and Avishai Mandelbaum. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing and Service Operations Management*, 5(2):79–141, 2003.

[26] O. Garnett, A. Mandelbaum, and M. Reiman. Designing a call center with impatient customers. *Manufacturing & Service Operations Management*, 4(3):208–27, 2002.

[27] Anders Gorst-Rasmussen and Martin B. Hansen. Asymptotic inference for waiting times and patiences in queues with abandonment. *Communications in Statistics - Simulation and Computation*, 38(2):318–334, 2009.

[28] Linda Green and Peter Kolesar. The pointwise stationary approximation for queues with nonstationary arrivals. *Management Science*, 37(1):84–97, 1991.

[29] Linda Green, Peter Kolesar, and Anthony Svoronos. Some effects of nonstationarity on multiserver markovian queueing systems. *Operations Research*, 39(3):502–511 and 171403, 1991.

[30] Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes*. Oxford University Press, third edition, 2001.

[31] R.C. Grinold and K.T. Marshall. *Manpower planning models*. Elsevier Science Publishing Co Inc., US, 1977.

[32] S. Gulati and S.A. Malcolm. Call center scheduling technology evaluation using simulation. *Proceeding of the 2001 Winter Simulation Conference (Cat. No.01CH37304)*, 2:1438–1442 vol.2, 2001.

[33] I. Gurvich, M. Armony, and A. Mandelbaum. Service-level differentiation in call centers with fully flexible servers. *Management Science*, 54:279–294, 2008.

[34] Shlomo Halfin and Ward Whitt. Heavy-traffic limits for queues with many exponential servers. *Operations Research*, 29(3):567–588, 1981.

[35] J.M. Harrison and Assaf Zeevi. A method for staffing large call centers based on stochastic fluid models. *Manufacturing & Service Operations Management*, 7(1):20–36, 2005.

[36] T.-Y. Huang. Analysis and modeling of a threshold based priority queueing system. *Computer Communications*, 24(3-4):284–291, 2001.

[37] Villy Bæk Iversen. *Teletraffic Engineering and Network Planning*. COM Center, Technical University of Denmark, 2006.

[38] James R. Jackson. Some problems in queueing with dynamic priorities. *Nav. Res. Logistics Quart.*, 7:235–249, 1960.

[39] James R. Jackson. Queues with dynamic priority discipline. *Management Science*, 8(1):18–34 and 2627272, 1961.

[40] N.K. Jaiswal. *Priority Queues*. Academic Press, New York and Londons, 1968.

[41] A.J.E.M. Janssen, J.S.H. van Leeuwaarden, and Bert Zwart. Refining square root safety staffing by expanding erlang c. Working paper, Version of December 20, 2008.

[42] Otis B. Jennings, Avishai Mandelbaum, William A. Massey, and Ward Whitt. Server staffing to meet time-varying demand. *Management Science*, 42(10):1383–1394, 1996.

[43] Tania Jiménez and Ger Koole. Scaling and comparison of fluid limits of queues applied to call centers with time-varying parameters. *OR Spectrum*, 26(3):413–422, 2004.

[44] G. Jongbloed and G. Koole. Managing uncertainty in call centres using poisson mixtures. *Applied Stochastic Models in Business and Industry*, 17(4):307–318, 2001.

[45] Jack P.C. Kleijnen. *Design and Analysis of Simulation Experiments*. Springer, 2008.

[46] Leonard Kleinrock. A delay dependent queue discipline. *Naval Research Logistics Quarterly*, 11(4), 1964.

[47] Leonard Kleinrock and Roy P. Finkelstein. Time dependent priority queues. *Operations Research*, 15(1):104–116, 1967.

[48] G. Koole, A. Pot, and J. Talim. Routing heuristics for multi-skill call centers. *Simulation Conference, 2003. Proceedings of the 2003 Winter*, 2:1813–1816, 2003.

[49] Ger Koole. Redefining the service level in call centers. Working paper, available from http://www.math.vu.nl/ koole/research/.

[50] Ger Koole and Avishai Mandelbaum. Queueing models of call centers: An introduction. *Annals of Operations Research*, 113(4):41–59, 2002.

[51] Ger Koole and Auke Pot. An overview of routing and staffing algorithms in multi-skill customer contact centers. Submitted for publication, 2006, available from http://www.math.vu.nl/ koole/research/.

[52] G.M. Koole and J. Talim. Exponential approximation of multi-skill call centers architecture. *Proc. QNETs 2000*, 23:1–10, 2000.

[53] Leon-Garcia and Widjaja. *Communication Networks*. McGraw Hill Higher Education, second edition, 2001.

[54] Liqiang Liu and Vidyadhar G. Kulkarni. Explicit solutions for the steady state distributions in m/ph/1 queues with workload dependent balking. *Queueing Systems*, 52(4):251–260, 2006.

[55] A. Mandelbaum, W.A. Massey, M.I. Reiman, and B. Rider. Time varying multiserver queues with abandonment and retrials. *Teletraffic Engineering in a Competitive World. Proceedings of the International Teletraffic Congress - ITC-16. Vol.3a*, 1:355–64 vol.1, 1999.

[56] T.A. Mazzuchi and R.B. Wallace. Analyzing skill-based routing call centers using discrete-event simulation and design experiment. *Simulation Conference, 2004. Proceedings of the 2004 Winter*, 2:1812–1820 vol.2, 2004.

[57] V. Mehrotra and J. Fama. Call center simulation modeling: methods, challenges, and opportunities. *Simulation Conference, 2003. Proceedings of the 2003 Winter*, 1:135–143, 2003.

[58] Douglas C. Montgomery. *Design and Analysis of Experiments*. Wiley, 6th edition, 2008.

[59] Conny Palm. Inhomogeneous telephone traffic in full-availability groups. *Ericsson Technics*, (1), 1937.

[60] Conny Palm. Några undersökningar över väntetider vid telefonanläggningar. *Särtryck ur Tekn. Medd. från Kungl. Telegrafstyrelsen*, (7-9), 1937.

[61] Auke Pot. *Planning and Routing Algorithms for Multi-Skill Contact Centers*. PhD thesis, Vrije Universiteit, 2006.

[62] Auke Pot, Sandjai Bhulai, and Ger Koole. A simple staffing method for multiskill call centers. *Manufacturing and Service Operations Management*, 10(3):421–428, 2008.

[63] R. Saltzman and V. Mehrotra. A manager-friendly platform for simulation modeling and analysis of call center queueing systems. *Simulation Conference, 2004. Proceedings of the 2004 Winter*, 1, 2004.

[64] M. Segal. The operator-scheduling problem: A network-flow approach. *Operations Research*, 22(4):808–823, 1974.

[65] Haipeng Shen and Jianhua Z. Huang. Analysis of call centre arrival data using singular value decomposition. *Applied Stochastic Models in Business and Industry*, 21(3):251–263, 2005.

[66] Adam Smith. *An Inquiry into the Nature and Causes of the Wealth of Nations*. W. Strahan and T. Cadell, London, United Kingdom, 1776.

[67] Raik Stolletz and Stefan Helber. Performance analysis of an inbound call center with skills-based routing. *OR Spectrum*, 26(3):331–352, 2004.

[68] David Y. Sze. A queueing model for telephone operator staffing. *Operations Research*, 32(2):229–249, 1984.

[69] James W. Taylor. A comparison of univariate time series methods for forecasting intraday arrivals at a call center. *Management Science*, 54(2):253–265, 2008.

[70] R.B. Wallace and R.M. Saltzman. Comparing skill-based routing call center simulations using c programming and arena models. *Proceedings of the 2005 Winter Simulation Conference (IEEE Cat. No.05CH37732C)*, page 9 pp., 2005.

[71] R.B. Wallace and W. Whitt. A staffing algorithm for call centers with skill-based routing. *Manufacturing & Service Operations Management*, 7(4):276–294, 2005.

[72] W. Whitt. Engineering solution of a basic call-center model. *Management Science*, 51(2):221–235, 2005.

[73] Ward Whitt. Understanding the efficiency of multi-server service systems. *Management Science*, 38(5):708–723, 1992.

[74] Ward Whitt. Predicting queueing delays. *Management Science*, 45(6):870–888, 1999.

[75] Ward Whitt. Fluid models for multiserver queues with abandonments. *Operations Research*, 54(1):37–54, 2006.

[76] Ward Whitt. Staffing a call center with uncertain arrival rate and absenteeism. *Production and Operations Management*, 15(1):88–102, 2006.

[77] Bo Zhang, Johan S.H. van Leeuwaarden, and Bert Zwart. Refined square root staffing for call centers with impatient customers. To appear in Manufacturing & Service Operations Management.

[78] Ety Zohar, Avishai Mandelbaum, and Nahum Shimkin. Adaptive behavior of impatient customers in tele-queues: Theory and empirical support. *Management Science*, 48(4):566–583, 2002.

Appendices

APPENDIX A

Queues with waiting time dependent service

Queues with waiting time dependent service

R. Bekker[†], G.M. Koole[†], B.F. Nielsen[⋆], T.B. Nielsen[⋆]

[†]Dept. Mathematics
VU University Amsterdam
De Boelelaan 1081, 1081 HV, the Netherlands.

[⋆]Dept. Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, 2800 Kgs. Lyngby, Denmark.

Abstract

Motivated by service levels in terms of the waiting-time distribution seen in e.g. call centers, we consider two models for systems with a service discipline that depends on the waiting time. The first model deals with a single server that continuously adapts its service rate based on the waiting time of the first customer in line. In the second model, one queue is served by a primary server which is supplemented by a secondary server when the waiting of the first customer in line exceeds a threshold. Using level crossings for the waiting-time process of the first customer in line, we derive steady-state waiting-time distributions for both models. The results are illustrated with numerical examples.

Keywords: Waiting-time distribution; Adaptive service rate; Call centers; Contact centers; Queues; Deterministic threshold; Overflow; Level crossing.

1 Introduction

In service systems, the tail probability (or distribution function) of the waiting time of customers is one of the main service-level indicators. For example, in call centers the service level is generally characterized by the telephone service factor (TSF), i.e., the fraction of calls whose delay fall below a prespecified target. Typically, call centers use a 80-20 TSF meaning that 80% of the calls should be taken into service within 20 seconds, see [12]. Motivated by performance measures in terms of tail probabilities of waiting times, we consider queueing systems where the service mechanism is based on waiting times of customers. This type of control policy is commonly used in call centers [20], and indeed the authors have often encountered it in various forms when working with call centers. However, the literature on it is limited. In the traditional queueing literature, routing and control are commonly based on the number of customers present.

The main goal of this paper is to find the steady-state waiting-time distribution for queueing systems where the service characteristics depend on the waiting time of the first customer in line. This type of service control seems to be new in the queueing literature, despite its widespread use in the industry. In the sequel we use FIL as an abbreviation of first customer in line.

We consider two Markovian queueing models: (i) single-server queues with FIL waiting-time dependent service speed and (ii) a queue with two heterogeneous servers, where the secondary server is only activated as soon as the FIL waiting time exceeds some target level. For both models, the analysis is based on the waiting process of the first customer in line (FIL-process). Using level crossings, we find the steady-state distribution of the FIL-process and derive the waiting-time distribution as a corollary.

First, in Section 2, we study the single-server model, where the service speed can be continuously adapted based on the waiting time of the first customer in line. This model is related to the study of dams and queueing systems with workload-dependent service rates, see e.g. [4], [5], [16] or [24]. The difference is that the service speed here depends on the waiting time instead of the amount of work present.

Second, in Section 3, we consider a system with a single queue and two heterogeneous servers, where the secondary server takes the first customer in line into service as soon as his waiting time exceeds some threshold. The primary motivation for this model stems from routing mechanisms in call centers with operators in front and back offices. Typically, the only task of operators in the front office would be to answer calls whereas operators in the back office would have other assignments and only answer calls under high load. A common problem is then how to meet the service level agreements while keeping the disturbance of the back office operators to a minimum, see [12] and references therein. Overflow problems are in general difficult to analyze, see [11], because the overflow traffic is not Poisson; the deterministic threshold of this model only adds to this. We believe though that the model is of independent interest and has its applications in other areas where the service level involves the (tail) distribution of the waiting time, as in, e.g., telecommunication and production systems.

Related to the heterogeneous-servers model above is the slow-server problem, see [18], [19], [23] and [25]. In the slow-server model, a single queue is served by two heterogeneous servers with service rates μ_1 and μ_2, where $\mu_1 > \mu_2$. In [23], the author gives qualitative and explicit quantitative results on when to maintain or discard the slow server. In the models of [18] and [19], customers can be assigned to one of the servers depending on the number of customers present. There it was shown that the fast server should always be used and that the slow server should only be used if the number of customers exceeds some threshold. This result was derived for an infinite waiting space. We note that in case of a finite queue length, the optimal policy is not necessarily of a threshold type, see [25].

The literature on queueing models where the service time process depends on the waiting time is limited. In [3], a system with time dependent overflow is approximated by a queue-length dependent overflow. Prioritization based on adding different constants to the waiting times of customers is introduced in [17] and referred to as dynamic prioritization. There are some studies of single-server queues where the service time depends on the waiting time experienced by the customer in service (instead of the first customer in line), see [6], [22] and [26]. Furthermore, in [7] the authors consider an M/M/2 queue where non-waiting customers receive a different rate of service than customers who first wait in line. Their analysis is based on the "system point method" [8], which is closely related to the level crossing equations [10] of Section 3.

Some numerical results are presented in Section 4. Conclusions and topics for further research can be found in Section 5.

2 Single-server queue

In this section we consider a single-server queue where the service speed depends on the waiting time of the first customer in line. In particular, we assume that customers arrive according to a Poisson process with rate λ and have exponentially distributed service requirements with mean $1/\mu$. The service discipline is assumed to be FIFO. Denote by W_t the waiting time of the first customer in the queue at time t, with the convention that $W_t = 0$ if the queue is empty. Also, let Y_t denote the number of customers in service at time t (thus $Y_t \in \{0, 1\}$). The service speed depends on the waiting time of the first customer in line and the service speed function is denoted by $r(\cdot)$. Let $r(0)$ be the service speed for state $(W_t, Y_t) = (0, 1)$ and 0 be the speed for state $(0, 0)$. For convenience, define $\rho_0 = \lambda/(\mu r(0))$. We assume that $r(\cdot)$ is strictly positive, left-continuous, and has a strictly positive right limit on $(0, \infty)$.

The process $\{(W_t, Y_t), t \geq 0\}$ can now be described as follows. Given that $W_{t_0} = w > 0$ and the next service completion is at time $t_1 > t_0$, the waiting-time process of the first customer in line during (t_0, t_1) behaves as $W_{t_0+t} = w + t$ and $Y_{t_0+t} = 1$. If S_w denotes the time until the next service completion, conditioned on the initial waiting time $w > 0$, then $\mathbb{P}(S_w > t) = \exp\left(-\mu \int_w^{w+t} r(y)\mathrm{d}y\right)$. At the moment of a service completion, the second customer in line (if there is any) becomes the first customer in line. Since the interarrival times between customers are exponentially distributed, we have

$$W_{t_1^+} = \left(W_{t_1^-} - A_\lambda\right)^+, \tag{1}$$

where $(x)^+ = \max\{x, 0\}$ and A_λ denotes an exponential random variable of rate λ.

It remains to specify the boundary cases of an empty queue. For $(0, Y_{t_0})$, the time until the next state transition has an exponential distribution with rate $\lambda + \mu r(0) Y_{t_0}$. For $(0, 1)$ the next state is $(0, 0)$ with probability $\mu r(0)/(\lambda + \mu r(0))$, or W_t starts to increase linearly as described above with probability $\lambda/(\lambda + \mu r(0))$. For $(0, 0)$, the next state is $(0, 1)$ with probability one.

Since the service requirements and interarrival times are exponentially distributed, the process $\{(W_t, Y_t), t \geq 0\}$ is a Markov process. Assuming that the system is stable (see [9, Corollary 4.2] for stability conditions), the process is regenerative and thus has a stationary distribution, see e.g. [2, Chapter VII]. Below, we determine the steady-state distribution of this process and derive from it the waiting-time distribution of an arbitrary customer. For this, we introduce the steady-state distribution of the FIL-process as $W^{\mathrm{FIL}}(x) = \lim_{t\to\infty} \mathbb{P}(W_t \leq x)$ and the corresponding density as $w^{\mathrm{FIL}}(x) = \mathrm{d}W^{\mathrm{FIL}}(x)/\mathrm{d}x$. For the atom in zero, Y_t is included in the notation as $W^{\mathrm{FIL}}(0, y) = \lim_{t\to\infty} \mathbb{P}(W_t = 0, Y_t = y)$.

Theorem 2.1 *We have $W^{\mathrm{FIL}}(0, 1) = \rho_0 W^{\mathrm{FIL}}(0, 0)$. The density of the FIL-process is*

$$w^{\mathrm{FIL}}(x) = \lambda\rho_0 W^{\mathrm{FIL}}(0, 0)\, \exp\left\{\int_0^x (\lambda - \mu r(y))\mathrm{d}y\right\},$$

where

$$W^{\mathrm{FIL}}(0, 0) = \left[1 + \rho_0 + \lambda\rho_0 \int_0^\infty \exp\left\{\int_0^x (\lambda - \mu r(y))\mathrm{d}y\right\} \mathrm{d}x\right]^{-1}.$$

It is instructive to derive the distribution of the FIL-process based on level crossing arguments. We refer to Remark 2.1 below for an alternative proof based on results in [5].

Proof For $x > 0$, using (1), the level crossing equations read

$$w^{\mathrm{FIL}}(x) = \int_{y=x}^{\infty} e^{-\lambda(y-x)} \mu r(y) w^{\mathrm{FIL}}(y) \mathrm{d}y. \tag{2}$$

The left-hand side corresponds to upcrossings of level x and the right-hand side corresponds to the long-run average number of downcrossings through level x. Observe that we have continuous upcrossings of waiting-time levels and downcrossings by jumps, where the jump sizes correspond to interarrival times between successive customers (in contrast to workloads in single-server queues). Taking derivatives on both sides of Equation (2) yields

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}x} w^{\mathrm{FIL}}(x) &= \lambda \left[\int_{y=x}^{\infty} e^{-\lambda(y-x)} \mu r(y) w^{\mathrm{FIL}}(y) \mathrm{d}y \right] - \mu r(x) w^{\mathrm{FIL}}(x) \\ &= (\lambda - \mu r(x)) w^{\mathrm{FIL}}(x), \end{aligned}$$

where the second step follows from (2). The solution of this first-order differential equation can be readily obtained as

$$w^{\mathrm{FIL}}(x) = C \exp \left\{ \int_0^x (\lambda - \mu r(y)) \mathrm{d}y \right\}. \tag{3}$$

Balancing the transitions between the interior part of the state space and the boundary part, we have

$$\lambda W^{\mathrm{FIL}}(0,1) = \int_0^{\infty} e^{-\lambda y} \mu r(y) w^{\mathrm{FIL}}(y) \mathrm{d}y.$$

Using the above and letting $x \downarrow 0$ in (2) yields $\lim_{x \downarrow 0} w^{\mathrm{FIL}}(x) = \lambda W^{\mathrm{FIL}}(0,1)$. Also, letting $x \downarrow 0$ in (3) determines the constant $C = \lim_{x \downarrow 0} w^{\mathrm{FIL}}(x) = \lambda W^{\mathrm{FIL}}(0,1)$.
Now, balancing the transitions between the two boundary states gives

$$\lambda W^{\mathrm{FIL}}(0,0) = \mu r(0) W^{\mathrm{FIL}}(0,1),$$

which enables us to determine the three constants in terms of $W^{\mathrm{FIL}}(0,0)$. Finally, using normalization, we have

$$W^{\mathrm{FIL}}(0,0) + W^{\mathrm{FIL}}(0,1) + \lambda W^{\mathrm{FIL}}(0,1) \int_0^{\infty} \exp \left\{ \int_0^x (\lambda - \mu r(y)) \mathrm{d}y \right\} \mathrm{d}x = 1.$$

Expressing $W^{\mathrm{FIL}}(0,1)$ in $W^{\mathrm{FIL}}(0,0)$ and solving for $W^{\mathrm{FIL}}(0,0)$ completes the proof. □

To determine the waiting time, we only need to consider the FIL-process at specific points in time. We introduce the waiting time an arbitrary customer experiences as W and the distribution of this as $W(x) = \mathbb{P}(W \leq x)$. Using PASTA, it is easy to see that the atom in zero of the waiting time is given by $\mathbb{P}(W = 0) = W^{\mathrm{FIL}}(0,0)$. In case of non-zero waiting times, the waiting times are given by the FIL-process embedded at epochs just before downward jumps.
Let $N_s(u,v)$ denote the number of customers taken into service during the interval $(u,v]$. Consider an infinitesimal interval $(t, t+h]$, $h > 0$. Then, $\mathbb{P}(W_t > x; N_s(t,t+h) = 1) = \int_x^{\infty} \mu r(y) h w^{\mathrm{FIL}}(y) \mathrm{d}y + o(h)$. Note that $\mathbb{P}(N_s(t,t+h) = 1)/h$ (for $h \to 0$) is the rate at

which customers are taken into service and, since every customer leaves the queue through the server and the system is stable, equals λ. Combining the above, we have

$$\begin{aligned}\mathbb{P}(W > x) &= \lim_{h\to 0} \mathbb{P}(W_t > x \mid N_s(t,t+h) = 1) \\ &= \lim_{h\to 0} \frac{\mathbb{P}(W_t > x; N_s(t,t+h) = 1)}{\mathbb{P}(N_s(t,t+h) = 1)} \\ &= \frac{1}{\lambda} \int_x^\infty \mu r(y) w^{\mathrm{FIL}}(y) \mathrm{d}y.\end{aligned}$$

The density of the steady-state waiting time, $w(x)$, can be obtained by differentiating the above:

Corollary 2.1 *For the steady-state waiting time, we have* $\mathbb{P}(W = 0) = W^{\mathrm{FIL}}(0,0)$ *and density*

$$w(x) = \frac{\mu r(x) w^{\mathrm{FIL}}(x)}{\lambda},$$

where $W^{\mathrm{FIL}}(0,0)$ *and* $w^{\mathrm{FIL}}(\cdot)$ *are given in Theorem 2.1.*

Remark 2.1 We note that the steady-state waiting time and FIL distributions take a similar form as the steady-state workload distribution of an M/M/1 queue with workload-dependent arrival and/or service rate, see e.g. [4], [16] or [2], p. 388. Also related is the elapsed waiting time process in the M/G/1 queue [21].
For positive values, the FIL-process is a special case of the model considered in [5], i.e., an on/off storage system with state-dependent rates restricted to up intervals. Applying [5, Theorem 1] combined with [5, Section 6] and taking (in the notation of [5]) $r_0(x) \equiv 1$, $\lambda_0(x) = \mu r(x)$ and $\lambda_1(x)/r_1(x) \equiv \lambda$ with $\lambda_1(x)$ and $r_1(x)$ tending to infinity, directly yields the FIL-density represented in (3). Furthermore, combining results on expected excursion times [5, Theorem 2] with standard renewal arguments provides the remaining constants. $\diamond$

Remark 2.2 For a renewal arrival process, the interior part of the state space can be straightforwardly adapted. In particular, W_t is still a Markov process for positive waiting times and the level crossing equation (2) then reads

$$w^{\mathrm{FIL}}(x) = \int_{y=x}^\infty \mu r(y) w^{\mathrm{FIL}}(y)(1 - A(y-x))\mathrm{d}y,$$

where $A(\cdot)$ is the interarrival-time distribution. Note that the above equation can be written as a Volterra integral equation of the second kind, see e.g. [27]. For the FIL process to be a Markov process, a supplementary variable is required to describe the elapsed interarrival time at the boundary of the state space, i.e., in case there is no customer in line. We note that Corollary 2.1 remains valid for a renewal process. $\diamond$

Example 2.1 The results become even more tractable in various special cases. Here, we consider the case of two service speeds determined by a threshold value of the waiting time of the first customer in the queue. Specifically, we assume that

$$r(x) = \begin{cases} r_1, & \text{for } 0 \le x \le K, \\ r_2, & \text{for } x > K. \end{cases}$$

This example may serve as an approximation for the case of two heterogeneous servers in Section 3, where the secondary server is only activated as soon as the FIL-process exceeds K.
Using Theorem 2.1 and Corollary 2.1, we may easily obtain the steady-state distribution of the FIL-process and the waiting time. Here, we present the atom in zero and the density of the waiting time. Let $\rho_i = \lambda/(\mu r_i)$, for $i = 1, 2$. After some straightforward calculations, we obtain

$$w(x) = \begin{cases} r_1\mu\rho_1 W(0) e^{-r_1\mu(1-\rho_1)x}, & \text{for } 0 < x \leq K, \\ r_2\mu\rho_1 W(0) e^{(r_2-r_1)\mu K} e^{-r_2\mu(1-\rho_2)x}, & \text{for } x > K, \end{cases}$$

where

$$W(0) = \left[\frac{1}{1-\rho_1} + \rho_1 e^{-r_1\mu(1-\rho_1)K} \left(\frac{1}{1-\rho_2} - \frac{1}{1-\rho_1} \right) \right]^{-1}.$$

3 Two-server queue

In this section we turn our attention to a system with two heterogeneous servers. As in Section 2 we use the concept of a FIL-process, where W_t denotes the waiting time of the first customer in line at time t. Again customers arrive to the queue according to a Poisson process with rate λ. A primary server handles jobs with exponentially distributed service times with mean $1/\mu_p$. A secondary server starts serving customers when W_t exceeds a threshold K. The service times at the secondary server are exponentially distributed with mean $1/\mu_s$. As in the one-server model of Section 2, the service discipline is FIFO and the servers will always complete a started job, i.e., the secondary server will finish an already started job even if W_t drops below K due to a service completion. In this section Y_t refers to the number of active secondary servers at time t, thus $Y_t \in \{0, 1\}$. For the system to be stable we assume $\lambda < \mu_p + \mu_s$. The described two-server system is depicted in Figure 1.

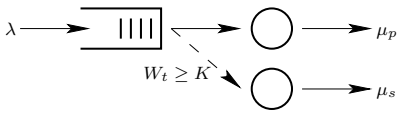


Figure 1: The queue is served by a primary server with rate μ_p which is supplemented by a secondary server with service rate μ_s, when the waiting time of the first in line, W_t, equals or exceeds K.

When dealing with the two-server setup, we introduce the steady-state joint distribution of the FIL-process as $W_i^{\text{FIL}}(x) = \lim_{t\to\infty} \mathbb{P}(W_t \leq x; Y_t = i)$. The joint steady-state density of the FIL-process is denoted $w_i^{\text{FIL}}(x)$.
A sample path of the FIL-process is shown in Figure 2. W_t increases linearly with time whenever a customer is in the queue. When the n'th customer enters service at time t, the waiting time of the first in line decreases with $\min(A_n, W_{t^-})$ from W_{t^-} to $W_{t^+} = \max(W_{t^-} - A_n, 0)$, where A_n is the exponentially distributed interarrival time with rate λ between customers n and $n+1$. Because both service times and interarrival times are exponentially distributed, the FIL-process is Markovian.

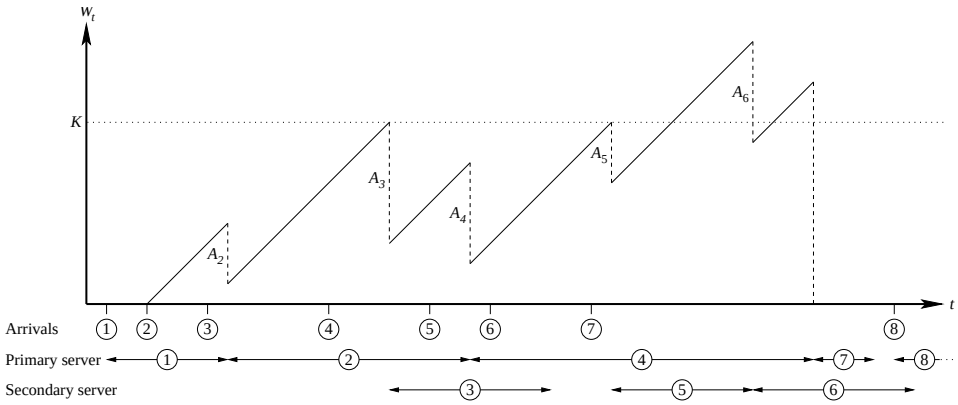


Figure 2: Elapsed waiting time of the first customer in line, W_t. The occupation of the servers are shown beneath the graph. Notice how W_t keeps increasing after customer #3 finishes service as the secondary server is not allowed to start a new service until the level K is reached.

The analysis of the system is based on the level crossing equations for the FIL-process. These are more involved, compared to those in Section 2, and are thus presented in Lemma 3.1. From this, the steady state distribution of the FIL-process is determined and given in Theorem 3.1.

Lemma 3.1 *We consider the level crossing equations with upcrossings of level x on the left-hand side and downcrossings on the right-hand for three different cases.*

(i) For $x < K$ and an active secondary server we have

$$\begin{aligned} w_1^{\mathrm{FIL}}(x) + \mu_s W_1^{\mathrm{FIL}}(x) = \mu_p \int_{y=x}^{\infty} e^{-\lambda(y-x)} w_1^{\mathrm{FIL}}(y)\mathrm{d}y \\ + \mu_s \int_{y=K}^{\infty} e^{-\lambda(y-x)} w_1^{\mathrm{FIL}}(y)\mathrm{d}y \\ + w_0^{\mathrm{FIL}}(K-) e^{-\lambda(K-x)}. \end{aligned}$$

(ii) For $x < K$ and an inactive secondary server

$$w_0^{\mathrm{FIL}}(x) = \mu_p \int_{y=x}^{K} e^{-\lambda(y-x)} w_0^{\mathrm{FIL}}(y)\mathrm{d}y + \mu_s W_1^{\mathrm{FIL}}(x).$$

(iii) For $x > K$ the secondary server will always be active

$$w_1^{\mathrm{FIL}}(x) = (\mu_p + \mu_s) \int_{y=x}^{\infty} e^{-\lambda(y-x)} w_1^{\mathrm{FIL}}(y)\mathrm{d}y.$$

Proof Only case (i) is dealt with in detail as it is the most complicated. The level crossing equations are obtained from setting up forward Kolmogorov equations. For case

(i) this becomes

$$\begin{aligned}
&\mathbb{P}(W_{t+\mathrm{d}t} \leq x + \mathrm{d}t; Y_{t+\mathrm{d}t} = 1) \\
&= (1 - \mu_p \mathrm{d}t - \mu_s \mathrm{d}t)\mathbb{P}(W_t \leq x; Y_t = 1) \\
&\quad + \mu_p \mathrm{d}t \mathbb{P}(W_t \leq x + A_n; Y_t = 1) \\
&\quad + \mu_s \mathrm{d}t \mathbb{P}(K < W_t \leq x + A_n; Y_t = 1) \\
&\quad + (1 - \mu_p \mathrm{d}t)\mathbb{P}(W_t \in [K - \mathrm{d}t, K]; W_t \leq x + A_n; Y_t = 0) + o(\mathrm{d}t).
\end{aligned}$$

Subtracting $\mathbb{P}(W_t \leq x + \mathrm{d}t; Y_t = 1)$ from both sides, dividing by $\mathrm{d}t$ and letting $\mathrm{d}t \to 0$ allows us to rewrite the term on the left side and the first term on the right side as derivatives with regard to t and x respectively. Moreover $\mathrm{d}t$ cancels from the rest of the terms except the last. Note that $\mu_p \mathbb{P}(W_t \in [K - \mathrm{d}t, K]; W_t \leq x + A_n; Y_t = 0) \to 0$ for $\mathrm{d}t \to 0$. Hence,

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{P}(W_t \leq x; Y_t = 1) \\
&= -\frac{\mathrm{d}}{\mathrm{d}x}\mathbb{P}(W_t \leq x; Y_t = 1) - (\mu_p + \mu_s)\mathbb{P}(W_t \leq x; Y_t = 1) \\
&\quad + \mu_p \mathbb{P}(W_t \leq x + A_n; Y_t = 1) + \mu_s \mathbb{P}(K < W_t \leq x + A_n; Y_t = 1) \\
&\quad + \lim_{\mathrm{d}t \to 0} \frac{\mathbb{P}(W_t \leq K; Y_t = 0) - \mathbb{P}(W_t \leq K - \mathrm{d}t; Y_t = 0)}{\mathrm{d}t} \cdot \mathbb{P}(A_n > K - x).
\end{aligned}$$

By letting $t \to \infty$, the left side of the expression tends to zero. The probabilities can be written in form of density and distribution functions, using convolution for the probabilities involving A_n; e.g. $\mathbb{P}(W_t \leq x + A_n; Y_t = 1) = W_1^{\mathrm{FIL}}(x) + \mathbb{P}(x < W_t \leq x + A_n, Y_t = 1) = W_1^{\mathrm{FIL}}(x) + \int_{y=x}^{\infty} e^{-\lambda(y-x)} w_1^{\mathrm{FIL}}(y)\mathrm{d}y$. Using $\lim_{\mathrm{d}t \to 0, t \to \infty} \left(\frac{\mathbb{P}(W_t \leq K; Y_t = 0) - \mathbb{P}(W_t \leq K - \mathrm{d}t; Y_t = 0)}{\mathrm{d}t} \right) = w_0^{\mathrm{FIL}}(K-)$, then leads to:

$$\begin{aligned}
0 = &- w_1^{\mathrm{FIL}}(x) - (\mu_p + \mu_s) W_1^{\mathrm{FIL}}(x) \\
&+ \mu_p \left(W_1^{\mathrm{FIL}}(x) + \int_{y=x}^{\infty} e^{-\lambda(y-x)} w_1^{\mathrm{FIL}}(y)\mathrm{d}y \right) + \mu_s \int_{y=K}^{\infty} e^{-\lambda(y-x)} w_1^{\mathrm{FIL}}(y)\mathrm{d}y \\
&+ w_0^{\mathrm{FIL}}(K-) e^{-\lambda(K-x)}.
\end{aligned}$$

Finally, the level crossing equation for case (i) can be obtained by simply rearranging the above terms.

We now turn to case (ii). Following an approach similar to the one for case (i), the level crossing equation can be found from the initial Kolmogorov equation

$$\begin{aligned}
\mathbb{P}\left(W_{t+\mathrm{d}t} \leq x + \mathrm{d}t; Y_{t+\mathrm{d}t} = 0\right) = &(1 - \mu_p \mathrm{d}t)\mathbb{P}(W_t \leq x; Y_t = 0) \\
&+ \mu_p \mathrm{d}t \mathbb{P}(W_t \leq x + A_n; Y_t = 0) \\
&+ \mu_s \mathrm{d}t \mathbb{P}(W_t \leq x; Y_t = 1) + o(\mathrm{d}t).
\end{aligned}$$

In case (iii) the Kolmogorov equation is of the following form

$$\begin{aligned}
\mathbb{P}(W_{t+\mathrm{d}t} \leq x + \mathrm{d}t; Y_{t+\mathrm{d}t} = 1) = &(1 - \mu_p \mathrm{d}t - \mu_s \mathrm{d}t)\mathbb{P}(W_t \leq x; Y_t = 1) \\
&+ (\mu_p + \mu_s)\mathrm{d}t \mathbb{P}(W_t \leq x + A_n; Y_t = 1) + o(\mathrm{d}t).
\end{aligned}$$

Again, using the same approach as for case (i), the level crossing equation of Lemma 3.1, case (iii), can be obtained. □

Theorem 3.1 *The density of the FIL-process, for $Y_t = 0$, is*

$$w_0^{\mathrm{FIL}}(x) = -c_1 e^{(\lambda-\mu_p)x} - r_1 c_3 e^{r_1 x} - r_2 c_4 e^{r_2 x}, \text{ for } 0 < x < K,$$

and, for $Y_t = 1$, it is

$$w_1^{\mathrm{FIL}}(x) = \begin{cases} r_1 c_3 e^{r_1 x} + r_2 c_4 e^{r_2 x}, & \text{for } 0 < x < K; \\ c_2 e^{(\lambda-\mu_p-\mu_s)x}, & \text{for } x > K, \end{cases}$$

with r_1, r_2 given by (6) and (7). The marginal density of the FIL-process for the two-server system becomes

$$w^{\mathrm{FIL}}(x) = \begin{cases} c_1 e^{(\lambda-\mu_p)x}, & \text{for } 0 < x < K; \\ c_2 e^{(\lambda-\mu_p-\mu_s)x}, & \text{for } x > K. \end{cases}$$

The constants c_i, $i \in \{1,\ 2,\ 3,\ 4\}$, are determined in Subsection 3.1.

Proof The densities of the FIL-process are found from the level crossing equations given in Lemma 3.1. The derivative with respect to x of the level crossing equation in case (i) becomes

$$\begin{aligned} w_1^{\mathrm{FIL}'}(x) + \mu_s W_1^{\mathrm{FIL}'}(x) = \lambda \Big[& \mu_p \int_{y=x}^{\infty} e^{-\lambda(y-x)} w_1^{\mathrm{FIL}}(y)\mathrm{d}y \\ & + \mu_s \int_{y=K}^{\infty} e^{-\lambda(y-x)} w_1^{\mathrm{FIL}}(y)\mathrm{d}y \\ & + w_0^{\mathrm{FIL}}(K-) e^{-\lambda(K-x)} \Big] \\ & - \mu_p w_1^{\mathrm{FIL}}(x), \end{aligned}$$

where the first and last term on the right-hand side of the above equation stem from the derivative of $\mu_p \int_{y=x}^{\infty} e^{-\lambda(y-x)} w_1^{\mathrm{FIL}}(y)\mathrm{d}y$. By rearranging and noting that the term inside the brackets equals $w_1^{\mathrm{FIL}}(x) + \mu_s W_1^{\mathrm{FIL}}(x)$, as given in the level crossing equation, we end up with a second-order differential equation:

$$W_1^{\mathrm{FIL}''}(x) + [\mu_p + \mu_s - \lambda]\, W_1^{\mathrm{FIL}'}(x) - \lambda\mu_s W_1^{\mathrm{FIL}}(x) = 0. \tag{4}$$

The general solution of (4) is of the form:

$$W_1^{\mathrm{FIL}}(x) = c_3 e^{r_1 x} + c_4 e^{r_2 x}, \tag{5}$$

where

$$r_1 = \frac{\lambda - (\mu_p + \mu_s) - \sqrt{(\mu_p + \mu_s - \lambda)^2 + 4\lambda\mu_s}}{2}, \tag{6}$$

$$r_2 = \frac{\lambda - (\mu_p + \mu_s) + \sqrt{(\mu_p + \mu_s - \lambda)^2 + 4\lambda\mu_s}}{2} \tag{7}$$

and c_3 and c_4 are constants. The derivative of (5) with respect to x yields the density, $w_1^{\mathrm{FIL}}(x)$, for $0 < x < K$, as given in Theorem 3.1.
The expressions for $w_0^{\mathrm{FIL}}(x)$ for $x < K$ and $w_1^{\mathrm{FIL}}(x)$ for $x > K$ can be found in the same way as the solution to the derivative of the level crossing equations in cases (ii) and (iii) of Lemma 3.1 respectively. Finally the marginal density of $w^{\mathrm{FIL}}(x)$ is found as the sum of $w_0^{\mathrm{FIL}}(x)$ and $w_1^{\mathrm{FIL}}(x)$. □

3.1 Constants and atoms

To fully describe the distribution of the FIL-process, the atoms in zero must be determined together with the constants in Theorem 3.1. The atoms, corresponding to the queue being empty, can be divided into four different boundary states; both servers are unoccupied (N), only the primary server is occupied (P), only the secondary server is occupied (S), and both servers are occupied (PS). The probabilities of being in these states are referred to as $W_{\mathrm{N}}^{\mathrm{FIL}}(0)$, $W_{\mathrm{P}}^{\mathrm{FIL}}(0)$, $W_{\mathrm{S}}^{\mathrm{FIL}}(0)$ and $W_{\mathrm{PS}}^{\mathrm{FIL}}(0)$, respectively.
Eight independent equations are needed to determine the eight constants; the probability of being in the four boundary states and the c_i's, $i \in \{1,\ 2,\ 3,\ 4\}$. Two equations follow directly from the boundary states in 0, as N and S can only be entered and left from other boundary states. Writing the rate out of the states on the left-hand side and the rate into the states on the right-hand side gives

$$\lambda W_{\mathrm{N}}^{\mathrm{FIL}}(0) = \mu_p W_{\mathrm{P}}^{\mathrm{FIL}}(0) + \mu_s W_{\mathrm{S}}^{\mathrm{FIL}}(0) \tag{8}$$

and

$$(\lambda + \mu_s) W_{\mathrm{S}}^{\mathrm{FIL}}(0) = \mu_p W_{\mathrm{PS}}^{\mathrm{FIL}}(0). \tag{9}$$

The rate out of P is $\lambda + \mu_p$ as this state can only be left by an arrival or a departure from the primary server. The state can be entered by an arrival in state N or a departure from the secondary server in state PS. P can also be entered from the FIL-process for non-zero W_t given that $Y_t = 0$ and $W_t < A_n$. This is represented by the second term on the right-hand side in (10).

$$(\lambda + \mu_p) W_{\mathrm{P}}^{\mathrm{FIL}}(0) = \lambda W_{\mathrm{N}}^{\mathrm{FIL}}(0) + \mu_p \int_{0^+}^{K} e^{-\lambda y} w_0^{\mathrm{FIL}}(y)\mathrm{d}y + \mu_s W_{\mathrm{PS}}^{\mathrm{FIL}}(0). \tag{10}$$

The balance equation for $W_{\mathrm{PS}}^{\mathrm{FIL}}(0)$ is found in the same way:

$$\begin{aligned}(\lambda + \mu_p + \mu_s) W_{\mathrm{PS}}^{\mathrm{FIL}}(0) = {} & \lambda W_{\mathrm{S}}^{\mathrm{FIL}}(0) + \mu_p \int_{0^+}^{\infty} e^{-\lambda y} w_1^{\mathrm{FIL}}(y)\mathrm{d}y \\ & + \mu_s \int_{K}^{\infty} e^{-\lambda y} w_1^{\mathrm{FIL}}(y)\mathrm{d}y + w_0^{\mathrm{FIL}}(K-) e^{-\lambda K}.\end{aligned} \tag{11}$$

Three more equations can be obtained by considering boundary conditions. By letting $x \downarrow 0$ in (5) we have

$$W_{\mathrm{S}}^{\mathrm{FIL}}(0) + W_{\mathrm{PS}}^{\mathrm{FIL}}(0) = c_3 + c_4. \tag{12}$$

Letting $x \uparrow K$ in the level crossing equation of case (ii) in Lemma 3.1 gives

$$\begin{aligned} w_0^{\mathrm{FIL}}(K-) &= \mu_s W_1^{\mathrm{FIL}}(K) \\ &= \mu_s \left[W_{\mathrm{S}}^{\mathrm{FIL}}(0) + W_{\mathrm{PS}}^{\mathrm{FIL}}(0) + \int_0^K w_1^{\mathrm{FIL}}(y)\mathrm{d}y \right], \end{aligned} \tag{13}$$

and the same limit in the level crossing equation of case (i) gives

$$\begin{aligned} w_1^{\mathrm{FIL}}(K-) + \mu_s W_1^{\mathrm{FIL}}(K) &= w_0^{\mathrm{FIL}}(K-) + (\mu_p + \mu_s) \int_{y=K}^{\infty} e^{-\lambda(y-K)} w_1^{\mathrm{FIL}}(y)\mathrm{d}y \\ &= w_0^{\mathrm{FIL}}(K-) + c_2 e^{(\lambda - \mu_p - \mu_s)K}. \end{aligned} \tag{14}$$

The final equation is obtained by normalization of the FIL-process:

$$1 = \int_0^K w_0^{\mathrm{FIL}}(y)\mathrm{d}y + \int_0^\infty w_1^{\mathrm{FIL}}(y)\mathrm{d}y + W_{\mathrm{N}}^{\mathrm{FIL}}(0) + W_{\mathrm{S}}^{\mathrm{FIL}}(0) + W_{\mathrm{P}}^{\mathrm{FIL}}(0) + W_{\mathrm{PS}}^{\mathrm{FIL}}(0). \quad (15)$$

The analytical expressions for the constants do not seem to give any additional insight into the problem. Solving the equations numerically is straightforward. We have shown that at most two of the equations can be mutually dependent and all numerical investigations point toward them being independent. Furthermore we argue that as long as the requirements for stability of the system are fulfilled, a unique solution to the equation array must exist and thus the equations must indeed be independent.

3.2 Waiting-time distribution

We now turn to the waiting-time distribution and use the same definition of this as in Section 2; $W(x) = \mathbb{P}(W \leq x)$, where W is the waiting time an arbitrary customer experiences. Observe that arriving customers are directly taken into service in case the queue is empty and the primary server is available. Using PASTA, it is easy to obtain the atom in zero of the waiting time:

$$\mathbb{P}(W = 0) = W_{\mathrm{N}}^{\mathrm{FIL}}(0) + W_{\mathrm{S}}^{\mathrm{FIL}}(0).$$

In case the waiting time is non-zero, the waiting time corresponds to the FIL-process at epochs right before downward jumps. Here, we again consider an infinitesimal interval $(t, t+h)$ and apply similar arguments as in Section 2. In particular, for $x \geq K$, we have

$$\mathbb{P}(W_t > x; N_s(t, t+h) = 1) = (\mu_p + \mu_s)h \int_x^\infty w_1^{\mathrm{FIL}}(y)\mathrm{d}y + o(h).$$

For $0 < x < K$, we have

$$\begin{aligned}\mathbb{P}(W_t > x; N_s(t, t+h) = 1) = \mu_p h \int_x^{K-h} w_0^{\mathrm{FIL}}(y)\mathrm{d}y + \mu_p h \int_x^K w_1^{\mathrm{FIL}}(y)\mathrm{d}y \\ + \int_{K-h}^K w_0^{\mathrm{FIL}}(y)\mathrm{d}y + (\mu_p + \mu_s)h \int_K^\infty w_1^{\mathrm{FIL}}(y)\mathrm{d}y + o(h).\end{aligned}$$

Note that $\int_{K-h}^K w_0^{\mathrm{FIL}}(y)\mathrm{d}y/h \to w_0^{\mathrm{FIL}}(K-)$, as $h \to 0$. Also, observe that $\mathbb{P}(N_s(t, t+h) = 1)/h$ (for $h \to 0$) is the rate at which customers are taken into service and, since every customer leaves the queue through the server and the system is stable, equals λ. Combining the above and using a similar conditioning as in Section 2, we obtain

$$\mathbb{P}(W > x) = \begin{cases} \frac{1}{\lambda}\Big[\mu_p \int_x^K (w_0^{\mathrm{FIL}}(y) + w_1^{\mathrm{FIL}}(y))\mathrm{d}y \\ \qquad + w_0^{\mathrm{FIL}}(K-) + (\mu_p + \mu_s)\int_K^\infty w_1^{\mathrm{FIL}}(y)\mathrm{d}y\Big], & \text{for } 0 \leq x < K, \\ \frac{\mu_p + \mu_s}{\lambda}\int_x^\infty w_1^{\mathrm{FIL}}(y)\mathrm{d}y, & \text{for } x \geq K. \end{cases} \quad (16)$$

From this, we obtain the density of the steady-state waiting time and the atom at K:

Corollary 3.1 *For the steady-state waiting time, we have two atoms*

$$\begin{aligned}\mathbb{P}(W=0) &= W_{\mathrm{N}}^{\mathrm{FIL}}(0)+W_{\mathrm{S}}^{\mathrm{FIL}}(0),\\ \mathbb{P}(W=K) &= \frac{w_0^{\mathrm{FIL}}(K-)}{\lambda},\end{aligned}$$

and density

$$w(x)=\begin{cases}\frac{\mu_p}{\lambda}c_1e^{(\lambda-\mu_p)x}, & \text{for } 0<x<K,\\ \frac{\mu_p+\mu_s}{\lambda}c_2e^{(\lambda-\mu_p-\mu_s)x}, & \text{for } x>K.\end{cases}$$

Remark 3.1 Note that the form of the steady-state waiting time density (and distribution) is closely related to the density in Example 2.1, i.e., the single-server model with two service speeds determined by a threshold on the FIL-process. In particular, the parameters r_i, $i=1,2$, and μ should be taken such that $r_1\mu=\mu_p$ and $r_2\mu=\mu_p+\mu_s$ (for instance, let $\mu=\mu_p$, $r_1=1$, and $r_2=1+\mu_s/\mu_p$). The main difference between the waiting-time distributions concerns the atom at K. ◇

4 Numerical results

To illustrate the difference in behavior of the waiting-time distribution for the one server system of Example 2.1 and the two-server system treated in Section 3, a few numerical results are shown in Figure 3. The parameters have been chosen such that the two cases are comparable.

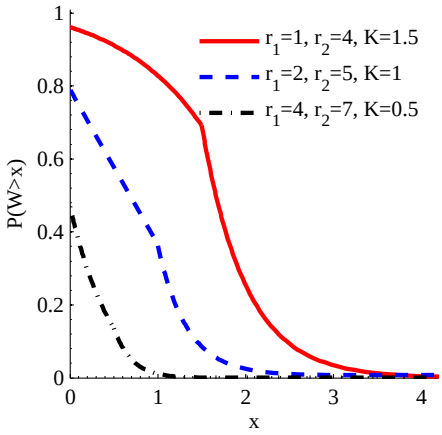
The waiting-time distributions in Figure 3b are found from Corollary 3.1 and the corresponding eight constants, found with Maple, are given in Table 1. It is seen how the relation between λ and μ_p governs the shape of the distribution for $x<K$; it is convex for $\lambda<\mu_p$, concave for $\lambda>\mu_p$ and a straight line for $\lambda=\mu_p$. Notable are also the atoms at K which are absent in the two-speed single server case of Figure 3a.

The somewhat better performance of the two-server model can be explained by the secondary server finishing an already started service when W_t drops below K, whereas the single server system of Example 2.1 will change the service speed to r_1 immediately.

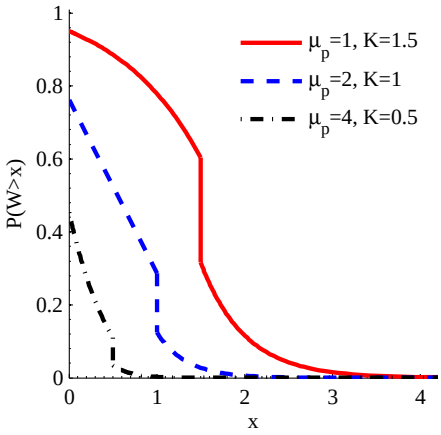
Table 1: Numerical results for common parameters $\lambda=2$, $\mu_s=3$.

	($\mu_p=1$, $K=1.5$)	($\mu_p=2$, $K=1.0$)	($\mu_p=4$, $K=0.5$)
W_{N}	0.0470	0.2298	0.5318
W_{P}	0.0860	0.2181	0.2559
W_{S}	0.0027	0.0078	0.0133
W_{PS}	0.0135	0.0195	0.0166
c_1	0.1990	0.4751	0.5451
c_2	6.3453	2.9956	0.6123
c_3	$-0.6401\cdot10^{-4}$	$-0.2673\cdot10^{-3}$	$-0.4749\cdot10^{-3}$
c_4	0.01626	0.0276	0.0304

In Figure 4 we compared the performance of the service mechism based on waiting times to the control based on queue lengths, since the latter is common in the queueing literature.



(a) Waiting-time distributions for the single server system in Example 2.1 with two service speeds determined by different parameter values of r_1, r_2, and K. Shared parameters: $\lambda = 2$ and $\mu = 1$.



(b) Waiting-time distribution for the two-server system of Section 3 for 3 different values of μ_p and K and shared parameters $\lambda = 2$ and $\mu_s = 3$.

Figure 3: Numerical comparison of the one and two-server models.

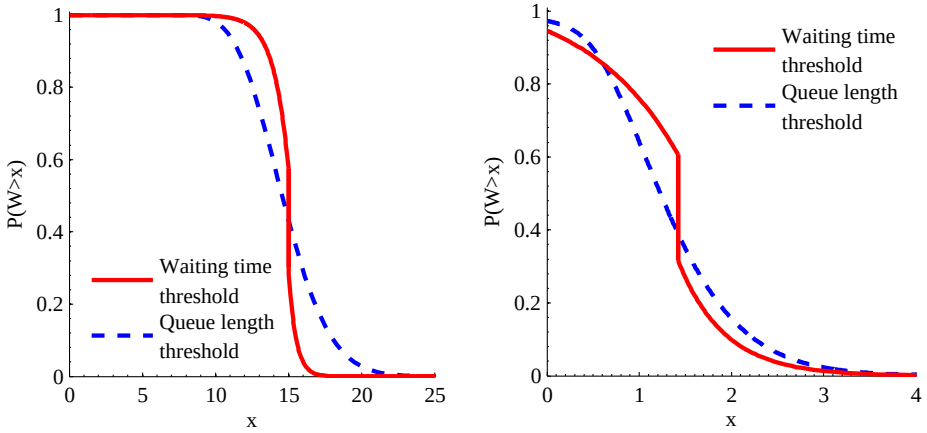
For the model with queue-length based control, the secondary server is only allowed to take customers into service when more than 30 and 3 customers, in Figures 4a and 4b, respectively, are waiting in the queue. These parameters have been chosen such that the resulting average waiting times are nearly identical for the two policies. The waiting-time distribution for the queue-length based threshold is found by taking the average of 50 simulations of 100.000 calls each. In this way the 95% confidence intervals become too narrow to display in the figure. It is seen that the waiting-time based threshold results in less variation of waiting times which is preferable as the objective is to have more control over the system. This reduction in variability of waiting times is accentuated for larger threshold value as displayed in Figure 4a. The figure illustrates the interesting, but not surprising, phenomenon of how the probability mass gathers around K for $\lambda > \mu_p$, K large and waiting-time based control.
Given the distribution of the waiting-time and FIL-process, most of the commonly used performance measures such as TSF are easily found. Other performance measures such as the utilization of the servers can be found as

$$\begin{aligned} a_p &= 1 - W_{\mathrm{N}}(0) - W_{\mathrm{S}}(0), \\ a_s &= 1 - W_{\mathrm{N}}(0) - W_{\mathrm{P}}(0) - \int_0^K w_0^{\mathrm{FIL}}(y)\mathrm{d}y, \end{aligned}$$

where a_p and a_s are the utilization of the primary and secondary server, respectively.

5 Conclusions and topics for further research

We have studied queueing systems where the provided service depends on the waiting time of the first customer in line. This type of control is commonly used in call centers and has



(a) Waiting-time distribution for large K, ($\lambda = 2$, $\mu_p = 1$, $\mu_s = 3$, $K = 15$) compared to a similar setup with queue-length based threshold of 30.

(b) Waiting-time distribution for small K, ($\lambda = 2$, $\mu_p = 1$, $\mu_s = 3$, $K = 1.42$) compared to a similar setup with queue-length based threshold of 3.

Figure 4: Waiting-time thresholds compared to queue-length thresholds.

mainly been motivated by a frequently used setup referred to as an "inverted V", see [1]. The main contribution is that we have shown ways to deal with systems where the service changes depending on the waiting time, which can be inherently difficult to deal with in particular in the case of fixed thresholds.

The first model of this paper deals with a single server that operates with a service speed depending on the waiting time of the first customer in line. We derived the waiting-time distribution of an arbitrary customer entering the system and showed how the model can be used for the threshold case.

The second model of this paper deals with a two-server setup where a secondary server supplements a primary server when the waiting time of the first in line exceeds a threshold. Again the waiting distribution of an arbitrary customer has been derived and numerical examples have been given. It was illustrated that a waiting-time based threshold is preferable to a queue-length based, when a high degree of control of the waiting times is desired. Also, The simplicity of the form of the solution for the waiting time given in Corollary 3.1 provides some useful insight.

In the model presented in Section 3, only one primary and one secondary server was considered. This is easily extended to a more general setup with multiple primary servers by introducing additional states for $W^{\mathrm{FIL}}(0)$ along with the four already used. The extra boundary states should describe the number of unoccupied servers. Analyzing a setup with multiple secondary servers would be much more difficult as the joint distribution of $w_i^{\mathrm{FIL}}(x)$ must be extended to include $i \in \{0,\ 1, ...,\ n\}$, where n is the number of secondary servers.

A related routing setup, often seen in call centers and used as a way to prioritize a group of customers over another, is the "N" design, see [12]. Also related are [13], [14] and [15]. The "N" design is basically an extension to the model of Section 3 where the secondary server also has a queue of its own, from which it receives jobs. Extending the model presented in

this paper to the "N" design, necessitates the use of a 2-dimensional FIL-process in order to keep track of the waiting time of the first customer in line in both queues.
There is still much to be done in relation to analysis of complex queueing systems such as those seen in call centers. Even though simulation may remain the dominant way of modelling these systems, it is indeed worth pursuing analytical approaches to gain insight not obtainable through simulation such as the result in Corollary 3.1.

References

[1] Armony, M. (2005). Dynamic routing in large-scale service systems with heterogeneous servers. *Queueing Systems* **51**, 287–329.

[2] Asmussen, S. (2003). *Applied Probability and Queues*, Second Edition. Springer, New York.

[3] Barth, W., M. Manitz, R. Stolletz (2009). Analysis of Two-Level Support Systems with Time-Dependent Overflow - A Banking Application. Forthcoming in *Production and Operations Management*.

[4] Bekker, R., S.C. Borst, O.J. Boxma, O. Kella (2004). Queues with workload-dependent arrival and service rates. *Queueing Systems* **46**, 537–556.

[5] Boxma, O., H. Kaspi, O. Kella, D. Perry (2005). On/off storage systems with state dependent input, output and switching rates. *Probability in the Engineering and Informational Sciences* **19**, 1–14.

[6] Boxma, O.J., M. Vlasiou (2007). On queues with service and interarrival times depending on waiting times. *Queueing Systems* **56**, 121–132.

[7] Brill, P.H., M.J.M. Posner (1981). A two server queue with nonwaiting customers receiving specialized service. *Management Science* **27**, 914–925.

[8] Brill, P.H., M.J.M. Posner (1981). The system point method in exponential queues: a level crossing approach. *Mathematics of Operations Research* **6**, 31–49.

[9] Browne, S., K. Sigman (1992). Work-modulated queues with applications to storage processes. *Journal of Applied Probability* **29**, 699–712.

[10] Cohen, J.W., M. Rubinovitch (1977). On level crossings and cycles in dam processes. *Mathematics of Operations Research* **2**, 297–310.

[11] Franx, G.J., G.M. Koole, S.A. Pot (2006). Approximating multi-skill blocking systems by hyperexponential decomposition. *Performance Evaluation* **63**, 799–824.

[12] Gans, N., G.M. Koole, A. Mandelbaum (2003). Telephone call centers: tutorial, review, and research prospects. *Manufacturing and Service Operations Management* **5**, 79–141.

[13] Gans, N., Y.-P. Zhou (2003). A call-routing problem with service-level constraints. *Operations Research* **51**, 255–271.

[14] Gans, N., Y.-P. Zhou (2007). Call-routing schemes for call-center outsourcing. *Manufacturing and Service Operations Management* **9**, 33–50.

[15] Gurvich, I., M. Armony, A. Mandelbaum (2008). Service-level differentiation in call centers with fully flexible servers. *Management Science* **54**, 279–294.

[16] Harrison, J.M., S.I. Resnick (1976). The stationary distribution and first exit probabilities of a storage process with general release rule. *Mathematics of Operations Research* **1**, 347–358.

[17] Jackson, J.R. (1960). Some problems in queueing with dynamic priorities. *Naval Research Logistics Quarterly* **7**, 235–249.

[18] Koole, G.M. (1995). A simple proof of the optimality of a threshold policy in a two-server queueing system. *Systems & Control Letters* **26**, 301–303.

[19] Lin, W., P.R. Kumar (1984). Optimal control of a queueing system with two heterogeneous servers. *IEEE Trans. Automat. Control* **29**, 696–703.

[20] Lucent Technologies (1999). *CentreVu Release 8 Advocate User Guide*. P.O. Box 4100, Crawfordsville, IN 47933, U.S.A.: Lucent Technologies.

[21] Perry, D., L. Benny (1989). Continuous Production/Inventory Model with Analogy to Certain Queueing and Dam Models. *Advances in Applied Probability* **21**, 123–141.

[22] Posner, M.J.M (1973). Single-server queues with service time dependent on waiting time. *Operations Research* **21**, 610–616.

[23] Rubinovitch, M. (1985). The slow server problem. *Journal of Applied Probability* **22**, 205–213.

[24] Scheinhardt, W.R.W., N. van Foreest, M. Mandjes (2005). Continuous feedback fluid queues. *Operations Research Letters* **33**, 551–559.

[25] Stockbridge, R.H. (1991). A martingale approach to the slow server problem. *Journal of Applied Probability* **28**, 480–486.

[26] Whitt, W. (1990). Queues with service times and interarrival times depending linearly and randomly upon waiting times. *Queueing Systems* **6**, 335–351.

[27] Zabreiko, P.P, A.I. Koshelev, M.A. Krasnosel'skii; transl. and ed. by T.O. Shaposhnikova, R.S. Anderssen and S.G. Mikhlin (1975). *Integral Equations: a Reference Text*. Monographs and textbooks on pure and applied mathematics. Noordhoff, Leiden.

A.1 Addendum

Independence of Equations

In this addendum to Appendix A, we examine the independence of the eight equations obtained for the two-server queue in the paper *Queues with waiting time dependent service*. It is shown that at most two of the equations can be dependent, but all numerical investigations point towards all eight equations being independent. The eight equations are re-listed below.

$$\lambda W_{\mathrm{N}}^{FIL}(0) = \mu_p W_{\mathrm{P}}^{FIL}(0) + \mu_s W_{\mathrm{S}}^{FIL}(0), \tag{A.1}$$

$$(\lambda+\mu_p) W_{\mathrm{P}}^{FIL}(0) = \lambda W_{\mathrm{N}}^{FIL}(0) + \mu_p \int_{0^+}^{K} e^{-\lambda y} w_0^{FIL}(y)\mathrm{d}y + \mu_s W_{\mathrm{PS}}^{FIL}(0), \tag{A.2}$$

$$(\lambda+\mu_s) W_{\mathrm{S}}^{FIL}(0) = \mu_p W_{\mathrm{PS}}^{FIL}(0), \tag{A.3}$$

$$(\lambda+\mu_p+\mu_s) W_{\mathrm{PS}}^{FIL}(0) = \lambda W_{\mathrm{S}}^{FIL}(0) + \mu_p \int_{0^+}^{\infty} e^{-\lambda y} w_1^{FIL}(y)\mathrm{d}y + \mu_s \int_{K}^{\infty} e^{-\lambda y} w_1^{FIL}(y)\mathrm{d}y + w_0^{FIL}(K)e^{-\lambda K}, \tag{A.4}$$

$$1 = \int_0^K w_0^{FIL}(y)\mathrm{d}y + \int_0^\infty w_1^{FIL}(y)\mathrm{d}y + W_{\mathrm{N}}^{FIL}(0) + W_{\mathrm{S}}^{FIL}(0) + W_{\mathrm{P}}^{FIL}(0) + W_{\mathrm{PS}}^{FIL}(0), \tag{A.5}$$

$$w_0^{FIL}(K^-) = \mu_s \left[W_{\mathrm{S}}^{FIL}(0) + W_{\mathrm{PS}}^{FIL}(0) + \int_0^K w_1^{FIL}(y)\mathrm{d}y \right] \tag{A.6}$$

$$w_1^{FIL}(K^-) + \mu_s W_1^{FIL}(K) = w_0^{FIL}(K) + c_2 e^{(\lambda-\mu_p-\mu_s)K}, \tag{A.7}$$

$$W_{\mathrm{S}}^{FIL}(0) + W_{\mathrm{PS}}^{FIL}(0) = c_3 + c_4, \tag{A.8}$$

where

$$w_0^{FIL}(x) = -c_1 e^{(\lambda-\mu_p)x} - r_1 c_3 e^{r_1 x} - r_2 c_4 e^{r_2 x}, \text{ for } 0 < x \le K,$$

$$w_1^{FIL}(x) = \begin{cases} r_1 c_3 e^{r_1 x} + r_2 c_4 e^{r_2 x}, & \text{for } 0 < x \le K; \\ c_2 e^{(\lambda-\mu_p-\mu_s)x}, & \text{for } x > K, \end{cases}$$

with r_1, r_2 given by

$$r_1 = \frac{\lambda - (\mu_p + \mu_s) - \sqrt{(\mu_p + \mu_s - \lambda)^2 + 4\lambda\mu_s}}{2},$$
$$r_2 = \frac{\lambda - (\mu_p + \mu_s) + \sqrt{(\mu_p + \mu_s - \lambda)^2 + 4\lambda\mu_s}}{2}.$$

Writing Equations (A.1)-(A.8) in matrix form as $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ where

$$\boldsymbol{x} = \begin{bmatrix} W_{\mathrm{N}}^{FIL}(0) \\ W_{\mathrm{P}}^{FIL}(0) \\ W_{\mathrm{S}}^{FIL}(0) \\ W_{\mathrm{PS}}^{FIL}(0) \\ c_3 \\ c_4 \\ c_1 \\ c_2 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

results in a rather complicated **A**-matrix, which should be non-singular for the eight equations to be independent. We introduce the following matrix operations:

- *SwitchRow*(a,b): Switches the position of rows a and b. For instance, *SwitchRow*(2,4) on rows numbered $[1, 2, 3, 4]$ yields $[1, 4, 3, 2]$.
- *InsertRow*(a,b): Inserts row a on row b's position and shifts the other rows as required. Example: *InsertRow*(2,4) on rows numbered $[1, 2, 3, 4]$ yields $[1, 4, 2, 3]$.
- *Eliminate*(a,b,c): Eliminates element (a, b) (row a, column b) of the matrix using element (c, b) and manipulates the remaining elements of row a accordingly. The procedure can formulated as:

  $$\mathbf{A}(a, .) = \mathbf{A}(a, .) - \frac{A(a, b)}{A(c, b)}\mathbf{A}(c, .).$$

  As an example, running *Eliminate*(2,1,3) on the matrix **B** yields **C** in Equation (A.9).

$$\mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & \frac{3}{7} & \frac{6}{7} \\ 7 & 8 & 9 \end{bmatrix} \tag{A.9}$$

For the purpose of showing independence of the eight equations, we can disregard the **b**-vector and only concentrate on showing non-singularity of the matrix **A**. This also means, we can interchange and shift columns without considering the effects this would otherwise have on **x**. We introduce *SwitchColumn*(a,b) and *InsertColumn*(a,b) in the same way as *SwitchRow*(a,b) and *InsertRow*(a,b) above. By running the following operations on **A**, we obtain relatively simple expressions for the first six columns of matrix.

$\mathbf{A}(2,.) = \mathbf{A}(2,.) - e^{\lambda K}\,(\mathbf{A}(6,.) + \mu_s\mathbf{A}(8,.))$, $\mathbf{A}(6,.) = \mathbf{A}(2,.) + \mu_s\mathbf{A}(8,.)$, *Eliminate*(2,7,6) , *Eliminate*(4,7,6) , *Eliminate*(5,7,6) , *Eliminate*(4,8,7) , *Eliminate*(5,8,7) , *SwitchColumn*(5,7) , *SwitchColumn*(6,8) , *Eliminate*(2,1,1) , *Eliminate*(5,1,1) , *Eliminate*(1,3,3) , *Eliminate*(2,3,3) , *Eliminate*(4,3,3) , *Eliminate*(5,3,3) , *Eliminate*(8,3,3) , *InsertColumn*(4,6) , *InsertColumn*(2,5) , *InsertRow*(3,2) , *InsertRow*(6,3) , *InsertRow*(7,4) , *SwitchRow*(6,7) , *SwitchColumn*(3,5) , *Eliminate*(5,3,6) , *SwitchColumn*(2,3) , *InsertRow*(6,2) , *SwitchColumn*(4,5) , *InsertRow*(4,6), $\mathbf{A}(6,.) = \mathbf{A}(6,.) + \mathbf{A}(7,.)$, *Eliminate*(7,6,8) , *SwitchRow*(6,8) $\mathbf{A}(8,.) = \mathbf{A}(8,.) - \mathbf{A}(7,.)$.

The matrix operations have been carried out in Maple. The resulting six leftmost columns of the modified matrix $\mathbf{A}_m$ become

$$\mathbf{A}_m(\,.\,,1..6) = \begin{bmatrix} -\lambda & \mu_p & 0 & 0 & 0 & \frac{\mu_p\mu_s}{\lambda+\mu_s} \\ 0 & \frac{\lambda+\mu_p}{\lambda} & 0 & 0 & 0 & \frac{\lambda+\mu_p}{\lambda} \\ 0 & 0 & -\lambda-\mu_s & 0 & 0 & \mu_p \\ 0 & 0 & 0 & -e^{K(\lambda-\mu_p)} & 0 & 0 \\ 0 & 0 & 0 & 0 & e^{K(\lambda-\mu_p-\mu_s)} & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{\lambda+\mu_p+\mu_s}{\lambda+\mu_s} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

This implies that, at most two of the eight equations can be dependent, as all the elements below the diagonal in the first six columns of **A** have the value zero and the diagonal elements cannot be zero for (positive) finite values of K, λ, μ_p, and μ_s.

Lower, right 2-by-2 matrix

Now, we only need to consider the lower right 2-by-2 part of $\mathbf{A}_m$ to show full independence. In theory this should a straightforward task, but unfortunately the expressions in the four elements have become rather complicated, due to the matrix manipulations done earlier. The most obvious way to show full

rank of the 2-by-2 matrix is to show the determinant cannot be 0. The four expressions are listed and simplified individually below.

$$\begin{aligned}
A_m(7,7) =& \frac{e^{K(r_1-\lambda)}\mu_p r_1 - \mu_p r_1 + r_1\mu_s e^{K(r_1-\lambda)} - \mu_s e^{K(r_1-\lambda)}\lambda}{r_1-\lambda} \\
& - \frac{\lambda^2 + 2\mu_s\lambda + \mu_s\mu_p + \mu_s^2}{\lambda+\mu_p+\mu_s} \\
=& \frac{\mu_p r_1\left(e^{K(r_1-\lambda)}-1\right) + \mu_s e^{K(r_1-\lambda)}(r_1-\lambda)}{r_1-\lambda} - \frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s} \\
=& \frac{\mu_p r_1\left(e^{K(r_1-\lambda)}-1\right)}{r_1-\lambda} + \mu_s e^{K(r_1-\lambda)} - \frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s} \\
=& \frac{(r_1(\mu_p+\mu_s)-\lambda\mu_s)e^{K(r_1-\lambda)} - \mu_p r_1}{r_1-\lambda} - \frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s} \\
=& \left(\frac{r_1\mu_p}{r_1-\lambda}+\mu_s\right)e^{K(r_1-\lambda)} - \frac{\mu_p r_1}{r_1-\lambda} - \frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s},
\end{aligned}$$

$$\begin{aligned}
A_m(7,8) =& \frac{e^{K(r_2-\lambda)}\mu_p r_2 - \mu_p r_2 + r_2\mu_s e^{K(r_2-\lambda)} - \mu_s e^{K(r_2-\lambda)}\lambda}{r_2-\lambda} \\
& - \frac{\lambda^2 + 2\mu_s\lambda + \mu_s\mu_p + \mu_s^2}{\lambda+\mu_p+\mu_s} \\
=& \frac{\mu_p r_2\left(e^{K(r_2-\lambda)}-1\right) + \mu_s e^{K(r_2-\lambda)}(r_2-\lambda)}{r_2-\lambda} - \frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s} \\
=& \frac{\mu_p r_2\left(e^{K(r_2-\lambda)}-1\right)}{r_2-\lambda} + \mu_s e^{K(r_2-\lambda)} - \frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s} \\
=& \frac{(r_2(\mu_p+\mu_s)-\lambda\mu_s)e^{K(r_2-\lambda)} - \mu_p r_2}{r_2-\lambda} - \frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s} \\
=& \left(\frac{r_2\mu_p}{r_2-\lambda}+\mu_s\right)e^{K(r_2-\lambda)} - \frac{\mu_p r_2}{r_2-\lambda} - \frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s},
\end{aligned}$$

$$
\begin{aligned}
&A_m(8,7)\\
&= \frac{1}{(r_1+\mu_p-\lambda)(\mu_p^2-\lambda^2)}\Big(\mu_s\mu_p\lambda^2-\mu_s\mu_p^3+\mu_s\mu_p^3e^{K(r_1+\mu_p-\lambda)}-\lambda^2\mu_se^{r_1K}\mu_p\\
&+\frac{\lambda^2(-r_1\mu_p+e^{r_1K}r_1\mu_p-e^{r_1K}r_1\lambda-r_1\mu_s-\mu_se^{r_1K}\lambda+r_1\lambda-r_1^2+\mu_se^{r_1K}r_1+\mu_s\lambda+r_1^2e^{r_1K})\mu_p}{r_1+\mu_p-\lambda}\\
&+\frac{\lambda^2(-r_1\mu_p+e^{r_1K}r_1\mu_p-e^{r_1K}r_1\lambda-r_1\mu_s-\mu_se^{r_1K}\lambda+r_1\lambda-r_1^2+\mu_se^{r_1K}r_1+\mu_s\lambda+r_1^2e^{r_1K})\mu_p^2}{r_1(r_1+\mu_p-\lambda)}\\
&-\frac{2\lambda^3(-r_1\mu_p+e^{r_1K}r_1\mu_p-e^{r_1K}r_1\lambda-r_1\mu_s-\mu_se^{r_1K}\lambda+r_1\lambda-r_1^2+\mu_se^{r_1K}r_1+\mu_s\lambda+r_1^2e^{r_1K})\mu_p}{r_1(r_1+\mu_p-\lambda)}\\
&-\frac{\lambda^3(-r_1\mu_p+e^{r_1K}r_1\mu_p-e^{r_1K}r_1\lambda-r_1\mu_s-\mu_se^{r_1K}\lambda+r_1\lambda-r_1^2+\mu_se^{r_1K}r_1+\mu_s\lambda+r_1^2e^{r_1K})}{r_1+\mu_p-\lambda}\\
&+\frac{\lambda^4(-r_1\mu_p+e^{r_1K}r_1\mu_p-e^{r_1K}r_1\lambda-r_1\mu_s-\mu_se^{r_1K}\lambda+r_1\lambda-r_1^2+\mu_se^{r_1K}r_1+\mu_s\lambda+r_1^2e^{r_1K})}{r_1(r_1+\mu_p-\lambda)}\Big)\\
&+\frac{\lambda^2+2\mu_s\lambda+\mu_s\mu_p+\mu_s^2}{\lambda+\mu_p+\mu_s}\\
&= \frac{1}{(r_1+\mu_p-\lambda)(\mu_p^2-\lambda^2)}\Big[\mu_s\mu_p\lambda^2-\mu_s\mu_p^3+\mu_s\mu_p^3e^{K(r_1+\mu_p-\lambda)}-\lambda^2\mu_se^{r_1K}\mu_p\\
&+\left(\lambda^2\mu_p-\lambda^3+\frac{\lambda^2\mu_p^2-2\lambda^3\mu_p+\lambda^4}{r_1}\right)\\
&\cdot\left(\frac{-r_1\mu_p+e^{r_1K}r_1\mu_p-e^{r_1K}r_1\lambda-r_1\mu_s-\mu_se^{r_1K}\lambda+r_1\lambda-r_1^2+\mu_se^{r_1K}r_1+\mu_s\lambda+r_1^2e^{r_1K}}{r_1+\mu_p-\lambda}\right)\Big]\\
&+\frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s}\\
&= \frac{1}{(r_1+\mu_p-\lambda)(\mu_p^2-\lambda^2)}\Big[\mu_p\mu_s\left(\lambda^2\left(1-e^{r_1K}\right)+\mu_p^2\left(e^{K(r_1+\mu_p-\lambda)}-1\right)\right)\\
&+\lambda^2\left(\mu_p-\lambda\right)\left(1+\frac{\mu_p-\lambda}{r_1}\right)\left(\frac{\left(1-e^{r_1K}\right)\left(\lambda\mu_s+r_1(\lambda-\mu_p-\mu_s-r_1)\right)}{r_1+\mu_p-\lambda}\right)\Big]\\
&+\frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s}\\
&= \frac{\mu_p\mu_s\left(\lambda^2\left(1-e^{r_1K}\right)+\mu_p^2\left(e^{K(r_1+\mu_p-\lambda)}-1\right)\right)}{(r_1+\mu_p-\lambda)(\mu_p^2-\lambda^2)}+\frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s}\\
&= -\frac{\mu_p\mu_s\lambda^2}{(r_1+\mu_p-\lambda)(\mu_p^2-\lambda^2)}e^{kr_1}+\frac{\mu_p^3\mu_s}{(r_1+\mu_p-\lambda)(\mu_p^2-\lambda^2)}e^{K(r_1+\mu_p-\lambda)}\\
&-\frac{\mu_p\mu_s}{r_1+\mu_p-\lambda}+\frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s},
\end{aligned}
$$

and

$$
\begin{aligned}
&A_m(8,8)\\
&= \frac{1}{(r_2+\mu_p-\lambda)(\mu_p^2-\lambda^2)}\Big(\mu_s\mu_p\lambda^2 - \mu_s\mu_p^3 + \mu_s\mu_p^3 e^{K(r_2+\mu_p-\lambda)} - \lambda^2\mu_s e^{r_2K}\mu_p\\
&+ \frac{\lambda^2(-r_2\mu_p + e^{r_2K}r_2\mu_p - e^{r_2K}r_2\lambda - r_2\mu_s - \mu_s e^{r_2K}\lambda + r_2\lambda - r_2^2 + \mu_s e^{r_2K}r_2 + \mu_s\lambda + r_2^2 e^{r_2K})\mu_p}{r_2+\mu_p-\lambda}\\
&+ \frac{\lambda^2(-r_2\mu_p + e^{r_2K}r_2\mu_p - e^{r_2K}r_2\lambda - r_2\mu_s - \mu_s e^{r_2K}\lambda + r_2\lambda - r_2^2 + \mu_s e^{r_2K}r_2 + \mu_s\lambda + r_2^2 e^{r_2K})\mu_p^2}{r_2(r_2+\mu_p-\lambda)}\\
&- \frac{2\lambda^3(-r_2\mu_p + e^{r_2K}r_2\mu_p - e^{r_2K}r_2\lambda - r_2\mu_s - \mu_s e^{r_2K}\lambda + r_2\lambda - r_2^2 + \mu_s e^{r_2K}r_2 + \mu_s\lambda + r_2^2 e^{r_2K})\mu_p}{r_2(r_2+\mu_p-\lambda)}\\
&- \frac{\lambda^3(-r_2\mu_p + e^{r_2K}r_2\mu_p - e^{r_2K}r_2\lambda - r_2\mu_s - \mu_s e^{r_2K}\lambda + r_2\lambda - r_2^2 + \mu_s e^{r_2K}r_2 + \mu_s\lambda + r_2^2 e^{r_2K})}{r_2+\mu_p-\lambda}\\
&+ \frac{\lambda^4(-r_2\mu_p + e^{r_2K}r_2\mu_p - e^{r_2K}r_2\lambda - r_2\mu_s - \mu_s e^{r_2K}\lambda + r_2\lambda - r_2^2 + \mu_s e^{r_2K}r_2 + \mu_s\lambda + r_2^2 e^{r_2K})}{r_2(r_2+\mu_p-\lambda)}\Big)\\
&+ \frac{\lambda^2 + 2\mu_s\lambda + \mu_s\mu_p + \mu_s^2}{\lambda+\mu_p+\mu_s}\\
&= -\frac{\mu_p\mu_s\lambda^2}{(r_2+\mu_p-\lambda)(\mu_p^2-\lambda^2)}e^{kr_2} + \frac{\mu_p^3\mu_s}{(r_2+\mu_p-\lambda)(\mu_p^2-\lambda^2)}e^{K(r_2+\mu_p-\lambda)}\\
&- \frac{\mu_p\mu_s}{r_2+\mu_p-\lambda} + \frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s}.
\end{aligned}
$$

We note the last part of each of the four expressions of $\mathbf{A}_m(7..8, 7..8)$ are identical. If we write the values as

$$
\begin{aligned}
A_m(7,7) &= a - x,\\
A_m(7,8) &= b - x,\\
A_m(8,7) &= c + x,\\
A_m(8,8) &= d + x,
\end{aligned}
$$

the determinant, D, of $A_m(7..8, 7..8)$ becomes:

$$
\begin{aligned}
D &= (a-x)(d+x) - (b-x)(c+x)\\
&= ad - bc + x(a - b + c - d),
\end{aligned}
\tag{A.10}
$$

where

$$
\begin{aligned}
a &= \left(\frac{r_1\mu_p}{r_1-\lambda}+\mu_s\right)e^{K(r_1-\lambda)} - \frac{\mu_p r_1}{r_1-\lambda},\\
b &= \left(\frac{r_2\mu_p}{r_2-\lambda}+\mu_s\right)e^{K(r_2-\lambda)} - \frac{\mu_p r_2}{r_2-\lambda},\\
c &= \frac{-\mu_p\mu_s\lambda^2}{(r_1+\mu_p-\lambda)(\mu_p^2-\lambda^2)}e^{kr_1} + \frac{\mu_p^3\mu_s}{(r_1+\mu_p-\lambda)(\mu_p^2-\lambda^2)}e^{K(r_1+\mu_p-\lambda)} - \frac{\mu_p\mu_s}{r_1+\mu_p-\lambda},\\
d &= \frac{-\mu_p\mu_s\lambda^2}{(r_2+\mu_p-\lambda)(\mu_p^2-\lambda^2)}e^{kr_2} + \frac{\mu_p^3\mu_s}{(r_2+\mu_p-\lambda)(\mu_p^2-\lambda^2)}e^{K(r_2+\mu_p-\lambda)} - \frac{\mu_p\mu_s}{r_2+\mu_p-\lambda},\\
x &= \frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s}.
\end{aligned}
$$

In order to make the above expressions appear nicer for further manipulations, we introduce:

$$
\begin{aligned}
P_1 &= \mu_s + r_1 = \lambda - \mu_p - r_2,\\
P_2 &= \mu_s + r_2 = \lambda - \mu_p - r_1,\\
R_1 &= \lambda - r_1 = r_2 + \mu_p + \mu_s,\\
R_2 &= \lambda - r_2 = r_1 + \mu_p + \mu_s
\end{aligned}
$$

thus

$$
\begin{aligned}
a &= \left(\mu_s - \frac{r_1\mu_p}{R_1}\right)e^{-kR_1} + \frac{\mu_p r_1}{R_1},\\
b &= \left(\mu_s - \frac{r_2\mu_p}{R_2}\right)e^{-kR_2} + \frac{\mu_p r_2}{R_2},\\
c &= \frac{\mu_p\mu_s\lambda^2}{P_2(\mu_p^2-\lambda^2)}e^{kr_1} - \frac{\mu_p^3\mu_s}{P_2(\mu_p^2-\lambda^2)}e^{-kP_2} + \frac{\mu_p\mu_s}{P_2},\\
d &= \frac{\mu_p\mu_s\lambda^2}{P_1(\mu_p^2-\lambda^2)}e^{kr_2} - \frac{\mu_p^3\mu_s}{P_1(\mu_p^2-\lambda^2)}e^{-kP_1} + \frac{\mu_p\mu_s}{P_1},\\
x &= \frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s}.
\end{aligned}
$$

Now Equation (A.10) is considered piece by piece using the relations given

at the end of this addendum. For the first part

$$\begin{aligned}
ad &= \left(\left(\mu_s - \frac{r_1\mu_p}{R_1}\right)e^{-kR_1} + \frac{\mu_p r_1}{R_1}\right)\left(\frac{\mu_p\mu_s\lambda^2}{P_1(\mu_p^2-\lambda^2)}e^{kr_2} - \frac{\mu_p^3\mu_s}{P_1(\mu_p^2-\lambda^2)}e^{-kP_1} + \frac{\mu_p\mu_s}{P_1}\right)\\
&= \left(\mu_s - \frac{r_1\mu_p}{R_1}\right)\frac{\mu_p\mu_s\lambda^2}{P_1(\mu_p^2-\lambda^2)}e^{-K(\mu_p+\mu_s)} - \left(\mu_s - \frac{r_1\mu_p}{R_1}\right)\frac{\mu_p^3\mu_s}{P_1(\mu_p^2-\lambda^2)}e^{-K(\lambda+\mu_s)}\\
&\quad + \left(\mu_s - \frac{r_1\mu_p}{R_1}\right)\frac{\mu_p\mu_s}{P_1}e^{-kR_1} + \frac{\mu_p r_1}{R_1}\frac{\mu_p\mu_s\lambda^2}{P_1(\mu_p^2-\lambda^2)}e^{kr_2} - \frac{\mu_p r_1}{R_1}\frac{\mu_p^3\mu_s}{P_1(\mu_p^2-\lambda^2)}e^{-kP_1}\\
&\quad + \frac{\mu_p r_1}{R_1}\frac{\mu_p\mu_s}{P_1}\\
&= \left(\frac{\mu_p\mu_s^2\lambda^2}{P_1(\mu_p^2-\lambda^2)} - \frac{\mu_p\mu_s\lambda^2}{\mu_p^2-\lambda^2}\right)e^{-K(\mu_p+\mu_s)} - \left(\frac{\mu_p^3\mu_s^2}{P_1(\mu_p^2-\lambda^2)} - \frac{\mu_p^3\mu_s}{\mu_p^2-\lambda^2}\right)e^{-K(\lambda+\mu_s)}\\
&\quad + \left(\frac{\mu_p\mu_s^2}{P_1} - \mu_p\mu_s\right)e^{-kR_1} + \frac{\mu_p\mu_s\lambda^2}{\mu_p^2-\lambda^2}e^{kr_2} - \frac{\mu_p^3\mu_s}{\mu_p^2-\lambda^2}e^{-kP_1} + \mu_p\mu_s,
\end{aligned}$$

and

$$\begin{aligned}
bc &= \left(\left(\mu_s - \frac{r_2\mu_p}{R_2}\right)e^{-kR_2} + \frac{\mu_p r_2}{R_2}\right)\left(\frac{\mu_p\mu_s\lambda^2}{P_2(\mu_p^2-\lambda^2)}e^{kr_1} - \frac{\mu_p^2}{P_2(\mu_p^2-\lambda^2)}e^{-kP_2} + \frac{\mu_p\mu_s}{P_2}\right)\\
&= \left(\mu_s - \frac{r_2\mu_p}{R_2}\right)\frac{\mu_p\mu_s\lambda^2}{P_2(\mu_p^2-\lambda^2)}e^{-K(\mu_p+\mu_s)} - \left(\mu_s - \frac{r_2\mu_p}{R_2}\right)\frac{\mu_p^3\mu_s}{P_2(\mu_p^2-\lambda^2)}e^{-K(\lambda+\mu_s)}\\
&\quad + \left(\mu_s - \frac{r_2\mu_p}{R_2}\right)\frac{\mu_p\mu_s}{P_2}e^{-kR_2} + \frac{\mu_p r_2}{R_2}\frac{\mu_p\mu_s\lambda^2}{P_2(\mu_p^2-\lambda^2)}e^{kr_1} - \frac{\mu_p r_2}{R_2}\frac{\mu_p^3\mu_s}{P_2(\mu_p^2-\lambda^2)}e^{-kP_2}\\
&\quad + \frac{\mu_p r_2}{R_2}\frac{\mu_p\mu_s}{P_2}\\
&= \left(\frac{\mu_p\mu_s^2\lambda^2}{P_2(\mu_p^2-\lambda^2)} - \frac{\mu_p\mu_s\lambda^2}{\mu_p^2-\lambda^2}\right)e^{-K(\mu_p+\mu_s)} - \left(\frac{\mu_p^3\mu_s^2}{P_2(\mu_p^2-\lambda^2)} - \frac{\mu_p^3\mu_s}{\mu_p^2-\lambda^2}\right)e^{-K(\lambda+\mu_s)}\\
&\quad + \left(\frac{\mu_p\mu_s^2}{P_2} - \mu_p\mu_s\right)e^{-kR_2} + \frac{\mu_p\mu_s\lambda^2}{\mu_p^2-\lambda^2}e^{kr_1} - \frac{\mu_p^3\mu_s}{\mu_p^2-\lambda^2}e^{-kP_2} + \mu_p\mu_s.
\end{aligned}$$

Gathering the former two expressions eliminates all non-exponential ele-

ments.

$$
\begin{aligned}
ad-bc = {} & \frac{\mu_p\mu_s^2\lambda^2}{\mu_p^2-\lambda^2}\left(\frac{1}{P_1}-\frac{1}{P_2}\right)e^{-K(\mu_p+\mu_s)} - \frac{\mu_p^3\mu_s^2}{\mu_p^2-\lambda^2}\left(\frac{1}{P_1}-\frac{1}{P_2}\right)e^{-K(\lambda+\mu_s)} \\
& + \left(\frac{\mu_p\mu_s^2}{P_1}-\mu_p\mu_s\right)e^{-kR_1} - \left(\frac{\mu_p\mu_s^2}{P_2}-\mu_p\mu_s\right)e^{-kR_2} \\
& + \frac{\mu_p\mu_s\lambda^2}{\mu_p^2-\lambda^2}e^{kr_2} - \frac{\mu_p\mu_s\lambda^2}{\mu_p^2-\lambda^2}e^{kr_1} - \frac{\mu_p^3\mu_s}{\mu_p^2-\lambda^2}e^{-kP_1} + \frac{\mu_p^3\mu_s}{\mu_p^2-\lambda^2}e^{-kP_2} \\
= {} & \frac{\mu_s\lambda^2(r_1-r_2)}{\mu_p^2-\lambda^2}e^{-K(\mu_p+\mu_s)} - \frac{\mu_p^2\mu_s(r_1-r_2)}{\mu_p^2-\lambda^2}e^{-K(\lambda+\mu_s)} \\
& - \frac{\mu_p\mu_s r_1}{\mu_s+r_1}e^{-kR_1} + \frac{\mu_p\mu_s r_2}{\mu_s+r_2}e^{-kR_2} \\
& + \frac{\mu_p\mu_s\lambda^2}{\mu_p^2-\lambda^2}e^{kr_2} - \frac{\mu_p\mu_s\lambda^2}{\mu_p^2-\lambda^2}e^{kr_1} - \frac{\mu_p^3\mu_s}{\mu_p^2-\lambda^2}e^{-kP_1} + \frac{\mu_p^3\mu_s}{\mu_p^2-\lambda^2}e^{-kP_2}
\end{aligned}
$$

The expression inside the parenthesis in Equation (A.10) is now considered. First part by part

$$
\begin{aligned}
a-b &= \left(\mu_s - \frac{r_1\mu_p}{R_1}\right)e^{-kR_1} + \frac{\mu_p r_1}{R_1} - \left(\mu_s - \frac{r_2\mu_p}{R_2}\right)e^{-kR_2} - \frac{\mu_p r_2}{R_2} \\
&= \left(\mu_s - \frac{r_1\mu_p}{R_1}\right)e^{-kR_1} - \left(\mu_s - \frac{r_2\mu_p}{R_2}\right)e^{-kR_2} + r_1 - r_2,
\end{aligned}
$$

and

$$
\begin{aligned}
c-d = {} & \frac{\mu_p\mu_s\lambda^2}{P_2(\mu_p^2-\lambda^2)}e^{kr_1} - \frac{\mu_p^3\mu_s}{P_2(\mu_p^2-\lambda^2)}e^{-kP_2} \\
& + \frac{\mu_p\mu_s}{P_2} - \frac{\mu_p\mu_s\lambda^2}{P_1(\mu_p^2-\lambda^2)}e^{kr_2} + \frac{\mu_p^3\mu_s}{P_1(\mu_p^2-\lambda^2)}e^{-kP_1} - \frac{\mu_p\mu_s}{P_1} \\
= {} & \frac{\mu_p\mu_s}{\mu_p^2-\lambda^2}\left(\frac{\lambda^2}{P_2}e^{kr_1} - \frac{\lambda^2}{P_1}e^{kr_2} - \frac{\mu_p^2}{P_2}e^{-kP_2} + \frac{\mu_p^2}{P_1}e^{-kP_1}\right) + r_2 - r_1,
\end{aligned}
$$

which together becomes

$$
\begin{aligned}
a-b+c-d = {} & \left(\mu_s - \frac{r_1\mu_p}{R_1}\right)e^{-kR_1} - \left(\mu_s - \frac{r_2\mu_p}{R_2}\right)e^{-kR_2} \\
& + \frac{\mu_p\mu_s}{\mu_p^2-\lambda^2}\left(\frac{\lambda^2}{P_2}e^{kr_1} - \frac{\lambda^2}{P_1}e^{kr_2} - \frac{\mu_p^2}{P_2}e^{-kP_2} + \frac{\mu_p^2}{P_1}e^{-kP_1}\right)
\end{aligned}
$$

The entire expression for the determinant becomes

$$
\begin{aligned}
D &= ad - bc + x(a - b + c - d) \\
&= \frac{\mu_s\lambda^2(r_1 - r_2)}{\mu_p^2 - \lambda^2} e^{-K(\mu_p+\mu_s)} - \frac{\mu_p^2\mu_s(r_1 - r_2)}{\mu_p^2 - \lambda^2} e^{-K(\lambda+\mu_s)} \\
&\quad - \frac{\mu_p\mu_s r_1}{\mu_s + r_1} e^{-kR_1} + \frac{\mu_p\mu_s r_2}{\mu_s + r_2} e^{-kR_2} \\
&\quad + \frac{\mu_p\mu_s\lambda^2}{\mu_p^2 - \lambda^2} e^{kr_2} - \frac{\mu_p\mu_s\lambda^2}{\mu_p^2 - \lambda^2} e^{kr_1} - \frac{\mu_p^3\mu_s}{\mu_p^2 - \lambda^2} e^{-kP_1} + \frac{\mu_p^3\mu_s}{\mu_p^2 - \lambda^2} e^{-kP_2} \\
&\quad + \frac{(\lambda + \mu_s)^2 + \mu_p\mu_s}{\lambda + \mu_p + \mu_s} \left[\left(\mu_s - \frac{r_1\mu_p}{R_1} \right) e^{-kR_1} - \left(\mu_s - \frac{r_2\mu_p}{R_2} \right) e^{-kR_2} \right. \\
&\quad \left. + \frac{\mu_p\mu_s}{\mu_p^2 - \lambda^2} \left(\frac{\lambda^2}{P_2} e^{kr_1} - \frac{\lambda^2}{P_1} e^{kr_2} - \frac{\mu_p^2}{P_2} e^{-kP_2} + \frac{\mu_p^2}{P_1} e^{-kP_1} \right) \right]
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\mu_s\lambda^2(r_1 - r_2)}{\mu_p^2 - \lambda^2} e^{-K(\mu_p+\mu_s)} - \frac{\mu_p^2\mu_s(r_1 - r_2)}{\mu_p^2 - \lambda^2} e^{-K(\lambda+\mu_s)} \\
&\quad + \left[\frac{(\lambda + \mu_s)^2 + \mu_p\mu_s}{\lambda + \mu_p + \mu_s} \left(\mu_s - \frac{r_1\mu_p}{R_1} \right) - \frac{\mu_p\mu_s r_1}{\mu_s + r_1} \right] e^{-kR_1} \\
&\quad - \left[\frac{(\lambda + \mu_s)^2 + \mu_p\mu_s}{\lambda + \mu_p + \mu_s} \left(\mu_s - \frac{r_2\mu_p}{R_2} \right) - \frac{\mu_p\mu_s r_2}{\mu_s + r_2} \right] e^{-kR_2} \\
&\quad + \frac{\mu_p\mu_s\lambda^2}{\mu_p^2 - \lambda^2} \left(\frac{x}{P_2} - 1 \right) e^{kr_1} - \frac{\mu_p\mu_s\lambda^2}{\mu_p^2 - \lambda^2} \left(\frac{x}{P_1} - 1 \right) e^{kr_2} \\
&\quad + \frac{\mu_p^3\mu_s}{\mu_p^2 - \lambda^2} \left(\frac{x}{P_1} - 1 \right) e^{-kP_1} - \frac{\mu_p^3\mu_s}{\mu_p^2 - \lambda^2} \left(\frac{x}{P_2} - 1 \right) e^{-kP_2}
\end{aligned}
$$

Finally, inserting the expressions for R_1, R_2, P_1 and P_2 yield

$$
\begin{aligned}
D = & \frac{\mu_s\lambda^2(r_1-r_2)}{\mu_p^2-\lambda^2}e^{-K(\mu_p+\mu_s)} \\
& - \frac{\mu_p^2\mu_s(r_1-r_2)}{\mu_p^2-\lambda^2}e^{-K(\lambda+\mu_s)} \\
& + \left[\frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s}\left(\frac{\lambda\mu_s-r_1(\mu_p+\mu_s)}{\lambda-r_1}\right)-\frac{\mu_p\mu_s r_1}{\mu_s+r_1}\right]e^{-K(\lambda-r_1)} \\
& - \left[\frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s}\left(\frac{\lambda\mu_s-r_2(\mu_p+\mu_s)}{\lambda-r_2}\right)-\frac{\mu_p\mu_s r_2}{\mu_s+r_2}\right]e^{-K(\lambda-r_2)} \\
& + \frac{\mu_p\mu_s\lambda^2}{\mu_p^2-\lambda^2}\left(\frac{\lambda^2+\lambda\mu_s}{(\lambda+\mu_p+\mu_s)(\mu_s+r_2)}-\frac{r_2}{\mu_s+r_2}\right)e^{kr_1} \\
& - \frac{\mu_p\mu_s\lambda^2}{\mu_p^2-\lambda^2}\left(\frac{\lambda^2+\lambda\mu_s}{(\lambda+\mu_p+\mu_s)(\mu_s+r_1)}-\frac{r_1}{\mu_s+r_1}\right)e^{kr_2} \\
& + \frac{\mu_p^3\mu_s}{\mu_p^2-\lambda^2}\left(\frac{\lambda^2+\lambda\mu_s}{(\lambda+\mu_p+\mu_s)(\mu_s+r_1)}-\frac{r_1}{\mu_s+r_1}\right)e^{-K(\mu_s+r_1)} \\
& - \frac{\mu_p^3\mu_s}{\mu_p^2-\lambda^2}\left(\frac{\lambda^2+\lambda\mu_s}{(\lambda+\mu_p+\mu_s)(\mu_s+r_2)}-\frac{r_2}{\mu_s+r_2}\right)e^{-K(\mu_s+r_2)} \qquad \text{(A.11)}
\end{aligned}
$$

In order to analyze the determinant further, we denote the eight lines in the expression (A.11) as $D(1)$ through $D(8)$.

Numerical investigations

Numerical evaluations of the determinant given in (A.11) indicate that the determinant is always negative for valid values of K, λ, μ_p and μ_s (all should be positive and finite). A more detailed numerical investigation of the individual elements of D is given in Table (A.1), where the rough magnitudes are shown for different parameter values.

A pattern is seen in the signs of the individual elements of the determinant by looking at the values in Table A.1. This pattern is shown in Table A.2

Analysis of the determinant

By looking at the values of the determinant elements in the numerical investigation, it is seen, it should be possible to show that the determinant is always

	$\lambda > \mu_p$			$\lambda < \mu_p$		
λ	1	1	1	1	1	1
μ_p	0.01	0.01	0.99	1.01	1.01	10
μ_s	1	10	0.02	0.01	10	0.01
$D(1)$	$2{\cdot}10^{-2}$	$4{\cdot}10^{-19}$	$2.4{\cdot}10^{-3}$	$-8.3{\cdot}10^{-4}$	$-2.0{\cdot}10^{-19}$	$-3.4{\cdot}10^{-24}$
$D(2)$	$-2{\cdot}10^{-8}$	$-4{\cdot}10^{-25}$	$-2.3{\cdot}10^{-3}$	$8.8{\cdot}10^{-4}$	$2.1{\cdot}10^{-19}$	$7.9{\cdot}10^{-4}$
$D(3)$	$1{\cdot}10^{-9}$	$8{\cdot}10^{-29}$	$2.0{\cdot}10^{-4}$	$2.4{\cdot}10^{-4}$	$3.0{\cdot}10^{-27}$	$3.0{\cdot}10^{-21}$
$D(4)$	$2{\cdot}10^{-0}$	$1{\cdot}10^{1}$	$1.5{\cdot}10^{-3}$	$7.7{\cdot}10^{-4}$	$7.3{\cdot}10^{-0}$	$9.2{\cdot}10^{-5}$
$D(5)$	$-6{\cdot}10^{-10}$	$-3{\cdot}10^{-30}$	$-1.2{\cdot}10^{0}$	$1.2{\cdot}10^{0}$	$1.4{\cdot}10^{-24}$	$3.3{\cdot}10^{-21}$
$D(6)$	$-4{\cdot}10^{2}$	$-1{\cdot}10^{4}$	$-9.8{\cdot}10^{0}$	$4.7{\cdot}10^{0}$	$4.7{\cdot}10^{5}$	$1.0{\cdot}10^{-3}$
$D(7)$	$4{\cdot}10^{-4}$	$1{\cdot}10^{-2}$	$9.1{\cdot}10^{0}$	$-5.0{\cdot}10^{0}$	$-5.1{\cdot}10^{5}$	$-2.4{\cdot}10^{17}$
$D(8)$	$5{\cdot}10^{-16}$	$2{\cdot}10^{-36}$	$1.1{\cdot}10^{0}$	$-1.3{\cdot}10^{0}$	$-1.5{\cdot}10^{-24}$	$-7.8{\cdot}10^{-1}$
D	$-4{\cdot}10^{2}$	$-1{\cdot}10^{4}$	$-7{\cdot}10^{-1}$	$-4{\cdot}10^{-1}$	$-3{\cdot}10^{4}$	$-2{\cdot}10^{17}$

Table A.1: Numerical evaluation of determinant elements ($K = 4.7$ used).

	$\lambda > \mu_p$	$\lambda < \mu_p$
$D(1)$	$+$	$-$
$D(2)$	$-$	$+$
$D(3)$	$+$	$+$
$D(4)$	$+$	$+$
$D(5)$	$-$	$+$
$D(6)$	$-$	$+$
$D(7)$	$+$	$-$
$D(8)$	$+$	$-$

Table A.2: Pattern of the signs of the determinant elements. It is seen how the sign of $D(3)$ and $D(4)$ remain positive, whereas the rest alternate with the relation of λ and μ_p.

negative. This can be done by showing the following inequalities are fulfilled,

$$D(1) + D(2) > 0, \tag{A.12}$$
$$D(3) + D(4) > 0, \tag{A.13}$$
$$D(5) + D(8) < 0, \tag{A.14}$$
$$D(6) + D(7) < 0, \tag{A.15}$$

together with

$$-1 < \frac{D(1) + D(2) + D(3) + D(4)}{D(6) + D(7)} < 0. \tag{A.16}$$

First, we show that $D(1) + D(2) > 0$:

$$\begin{aligned} D(1) + D(2) &= \frac{\mu_s \lambda^2 (r_1 - r_2)}{\mu_p^2 - \lambda^2} e^{-K(\mu_p + \mu_s)} - \frac{\mu_p^2 \mu_s (r_1 - r_2)}{\mu_p^2 - \lambda^2} e^{-K(\lambda + \mu_s)} \\ &= \frac{\mu_s (r_1 - r_2)}{\mu_p^2 - \lambda^2} e^{-K\mu_s} \left(\lambda^2 e^{-K\mu_p} - \mu_p^2 e^{-K\lambda} \right). \end{aligned}$$

The nominator of the fraction and the middle exponential term are always negative. For $\lambda > \mu_p$, the denominator is negative and the expression inside the parentheses is positive. When $\lambda < \mu_p$, both of these change their sign, thus, the whole expression is always positive.

$D(3) + D(4)$ is by far the least nice term because of the many r's involved in the expression. It appears to always be positive from the numerical results, this has not been verified analytically though:

$$\begin{aligned} D(3) + D(4) = &+ \left[\frac{(\lambda + \mu_s)^2 + \mu_p \mu_s}{\lambda + \mu_p + \mu_s} \left(\frac{\lambda \mu_s - r_1(\mu_p + \mu_s)}{\lambda - r_1} \right) - \frac{\mu_p \mu_s r_1}{\mu_s + r_1} \right] e^{-K(\lambda - r_1)} \\ &- \left[\frac{(\lambda + \mu_s)^2 + \mu_p \mu_s}{\lambda + \mu_p + \mu_s} \left(\frac{\lambda \mu_s - r_2(\mu_p + \mu_s)}{\lambda - r_2} \right) - \frac{\mu_p \mu_s r_2}{\mu_s + r_2} \right] e^{-K(\lambda - r_2)} \end{aligned}$$

Now we look at $D(5) + D(8)$, which seems to be negative judging from the numerical experiments.

$$\begin{aligned} D(5) + D(8) = &\frac{\mu_p \mu_s \lambda^2}{\mu_p^2 - \lambda^2} \left(\frac{\lambda^2 + \lambda \mu_s}{(\lambda + \mu_p + \mu_s)(\mu_s + r_2)} - \frac{r_2}{\mu_s + r_2} \right) e^{K r_1} \\ &- \frac{\mu_p^3 \mu_s}{\mu_p^2 - \lambda^2} \left(\frac{\lambda^2 + \lambda \mu_s}{(\lambda + \mu_p + \mu_s)(\mu_s + r_2)} - \frac{r_2}{\mu_s + r_2} \right) e^{-K(\mu_s + r_2)} \\ = &\frac{\mu_p \mu_s}{\mu_p^2 - \lambda^2} \frac{\lambda^2 + \lambda \mu_s - r_2(\lambda + \mu_p + \mu_s)}{(\lambda + \mu_p + \mu_s)(\mu_s + r_2)} \left(\lambda^2 e^{-K\mu_p} - \mu_p^2 e^{-K\lambda} \right). \end{aligned}$$

In the expression for $D(5) + D(8)$, the first fraction alternates with opposite sign compared to the last parentheses, with the relation between λ and μ_p, thus always creating a negative sign together. The nominator of the second fraction seems to take both positive and negative values, this needs further investigation though.

Finally, $D(6) + D(7)$ is considered.

$$\begin{aligned}
D(6)+D(7) =& -\frac{\mu_p\mu_s\lambda^2}{\mu_p^2-\lambda^2}\left(\frac{\lambda^2+\lambda\mu_s}{(\lambda+\mu_p+\mu_s)(\mu_s+r_1)}-\frac{r_1}{\mu_s+r_1}\right)e^{kr_2}\\
&+\frac{\mu_p^3\mu_s}{\mu_p^2-\lambda^2}\left(\frac{\lambda^2+\lambda\mu_s}{(\lambda+\mu_p+\mu_s)(\mu_s+r_1)}-\frac{r_1}{\mu_s+r_1}\right)e^{-K(\mu_s+r_1)}\\
=&\frac{\mu_p\mu_s}{\mu_p^2-\lambda^2}\frac{\lambda^2+\lambda\mu_s-r_1(\lambda+\mu_p+\mu_s)}{(\lambda+\mu_p+\mu_s)(\mu_s+r_1)}e^{-K(\mu_s+r_1)}\left(\mu_p^2-\lambda^2e^{K(\lambda-\mu_p)}\right)\\
=&\frac{\mu_p\mu_s}{\mu_p^2-\lambda^2}\frac{\lambda^2+\lambda\mu_s-r_1(\lambda+\mu_p+\mu_s)}{(\lambda+\mu_p+\mu_s)(\mu_s+r_1)}e^{-K(\mu_s+r_1-\lambda)}\left(\mu_p^2e^{-K\lambda}-\lambda^2e^{-K\mu_p}\right)
\end{aligned}$$

Here, the first fraction and the parentheses at the end always have the same sign, meaning they are positive together. The denominator of the second fraction is always negative, due to $r_1 < -\mu_s$, as shown at the end of this addendum. Thus, the term $D(6) + D(7)$ is always negative.

We have now shown the inequalities (A.12) and (A.15) hold. The inequalities (A.13) and (A.14) remain to be finally proved.

By showing Equation (A.16) holds, it should be possible to prove the elements $D(1)$, $D(2)$, $D(3)$, $D(4)$, $D(6)$, and $D(7)$ are less than zero together.

$$\begin{aligned}
&\frac{D(1)+D(2)+D(3)+D(4)}{D(6)+D(7)}\\
&=\left\{\frac{\mu_s(r_1-r_2)}{\mu_p^2-\lambda^2}e^{-K\mu_s}\left(\lambda^2e^{-K\mu_p}-\mu_p^2e^{-K\lambda}\right)\right.\\
&+\left[\frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s}\left(\frac{\lambda\mu_s-r_1(\mu_p+\mu_s)}{\lambda-r_1}\right)-\frac{\mu_p\mu_sr_1}{\mu_s+r_1}\right]e^{-K(\lambda-r_1)}\\
&\left.-\left[\frac{(\lambda+\mu_s)^2+\mu_p\mu_s}{\lambda+\mu_p+\mu_s}\left(\frac{\lambda\mu_s-r_2(\mu_p+\mu_s)}{\lambda-r_2}\right)-\frac{\mu_p\mu_sr_2}{\mu_s+r_2}\right]e^{-K(\lambda-r_2)}\right\}\\
&\Big/\left\{\frac{\mu_p\mu_s}{\mu_p^2-\lambda^2}\frac{\lambda^2+\lambda\mu_s-r_1(\lambda+\mu_p+\mu_s)}{(\lambda+\mu_p+\mu_s)(\mu_s+r_1)}e^{-K(\mu_s+r_1-\lambda)}\left(\mu_p^2e^{-K\lambda}-\lambda^2e^{-K\mu_p}\right)\right\}
\end{aligned}$$

It is of course possible to interchange r_1's and r_2's from the expressions at the end of this addendum, but it does not seem to simplify the expression.

To summarize, it has been shown that at least seven of the eight equations (A.1)-(A.8) are independent. Furthermore, a strong indication of all eight being independent has been given using numerical examples with different parameter values covering a representative set of cases.

Useful relations regarding the r's

A few relations used for the manipulations in the earlier sections are listed in this section.

$$r_1 + \mu_p - \lambda < 0$$

Proof

$$\begin{aligned}
\mu_p - \lambda + \frac{1}{2}\left(\lambda - \mu_p - \mu_s - \sqrt{(-\lambda + \mu_p + \mu_s)^2 + 4\lambda\mu_s}\right) &< 0 \\
\frac{1}{2}(-\lambda + \mu_p - \mu_s) &< \frac{1}{2}\sqrt{(-\lambda + \mu_p + \mu_s)^2 + 4\lambda\mu_s} \\
-\lambda + \mu_p - \mu_s &< \sqrt{(-\lambda + \mu_p + \mu_s)^2 + 4\lambda\mu_s}
\end{aligned}$$

$$r_2 + \mu_p - \lambda > 0$$

Proof

$$\begin{aligned}
\mu_p - \lambda + \frac{1}{2}\left(\lambda - \mu_p - \mu_s + \sqrt{(-\lambda + \mu_p + \mu_s)^2 + 4\lambda\mu_s}\right) &> 0 \\
-\lambda + \mu_p - \mu_s + \sqrt{(-\lambda + \mu_p + \mu_s)^2 + 4\lambda\mu_s} &> 0 \\
\sqrt{(-\lambda + \mu_p + \mu_s)^2 + 4\lambda\mu_s} &> \lambda - \mu_p + \mu_s \\
(-\lambda + \mu_p + \mu_s)^2 + 4\lambda\mu_s &> \lambda^2 + \mu_p^2 + \mu_s^2 - 2\lambda\mu_p + 2\lambda\mu_s - 2\mu_p\mu_s \\
\lambda^2 + \mu_p^2 + \mu_s^2 - 2\lambda\mu_p - 2\lambda\mu_s + 2\mu_p\mu_s + 4\lambda\mu_s &> \lambda^2 + \mu_p^2 + \mu_s^2 - 2\lambda\mu_p + 2\lambda\mu_s - 2\mu_p\mu_s \\
4\mu_p\mu_s &> 0
\end{aligned}$$

$$r_1 < 0$$

Proof

$$\begin{aligned}
r_2 + \mu_p - \lambda &> 0 \\
\lambda - \mu_p - \mu_s - r_1 + \mu_p + \mu_p - \lambda &> 0 \\
-\mu_s &> r_1
\end{aligned}$$

$$0 < r_2 < \lambda$$

$$
\begin{aligned}
r_1^2 &= (\lambda - \mu_p - \mu_s) r_1 + \lambda \mu_s \\
r_2^2 &= (\lambda - \mu_p - \mu_s) r_2 + \lambda \mu_s \\
r_1 r_2 &= -\lambda \mu_s \\
r_1 + r_2 &= \lambda - \mu_p - \mu_s
\end{aligned}
$$

$$
R_1 R_2 = \lambda \mu_p
$$

$$
\begin{aligned}
R_1 P_1 &= (\lambda - r_1)(\mu_s + r_1) \\
&= -r_1^2 + (\lambda - \mu_s) r_1 + \lambda \mu_s \\
&= -(\lambda - \mu_p - \mu_s) r_1 - \lambda \mu_s + (\lambda - \mu_s) r_1 + \lambda \mu_s \\
&= \mu_p r_1
\end{aligned}
$$

$$
\begin{aligned}
R_2 P_2 &= (\lambda - r_2)(\mu_s + r_2) \\
&= -r_2^2 + (\lambda - \mu_s) r_2 + \lambda \mu_s \\
&= -(\lambda - \mu_p - \mu_s) r_2 - \lambda \mu_s + (\lambda - \mu_s) r_2 + \lambda \mu_s \\
&= \mu_p r_2
\end{aligned}
$$

$$
\begin{aligned}
&\frac{1}{P_1} - \frac{1}{P_2} \\
&= \frac{P_2 - P_1}{P_1 P_2} \\
&= \frac{\mu_s + r_2 - \mu_s - r_1}{(\mu_s + r_1)(\mu_s + r_2)} \\
&= \frac{r_2 - r_1}{\mu_s^2 + \mu_s (r_1 + r_2) + r_1 r_2} \\
&= \frac{r_2 - r_1}{\mu_s^2 + \mu_s (\lambda - \mu_p - \mu_s) + \frac{1}{4}((\lambda - \mu_p - \mu_s)^2 - (\lambda - \mu_p - \mu_s)^2 - 4\lambda \mu_s)} \\
&= \frac{r_1 - r_2}{\mu_p \mu_s}
\end{aligned}
$$

APPENDIX B

Waiting time dependent multi-server priority queues

Waiting time dependent multi-server priority queues

G.M. Koole[†], B.F. Nielsen[⋆], T.B. Nielsen[⋆]

[†]Dept. Mathematics
VU University Amsterdam
De Boelelaan 1081, 1081 HV, the Netherlands.

[⋆]Dept. Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, 2800 Kgs. Lyngby, Denmark.

May 2, 2010

Abstract

We introduce a new approach to modelling queueing systems where the priority or the routing of customers depends on the waiting time of the first customer in line. The approach uses the waiting time of the first customer in line as the primary variable and we obtain waiting time distributions for complex systems, such as the N-design routing scheme widely used in e.g. call centers and systems with dynamic priorities.

Keywords: Waiting time distribution; Call centers; Priority queues; Deterministic threshold; Erlang distribution; Dynamic priority; Due-date.

1 Introduction

The traditional approach to modelling queueing systems developed by Erlang, Engset, Fry and Molina has been to look at the number of customers in queue, see [8] for original references. In this paper we introduce an alternative way of modelling queueing systems useful for priority rules often used in real scenarios.

Prioritization of customers in real queueing systems is often implemented as a function of the waiting time of the customer first in line (FIL). This is for example used in call centers to route calls between agent pools in order to meet service levels of different customer classes [6]. The service levels are usually characterized by the telephone service factor (TSF), i.e. the fraction of calls answered within a certain time. The use of TSF thus motivates the examination of waiting time distributions rather than just average values. Prioritization based on waiting times is also used in the health care sector for e.g. operating room scheduling [3].

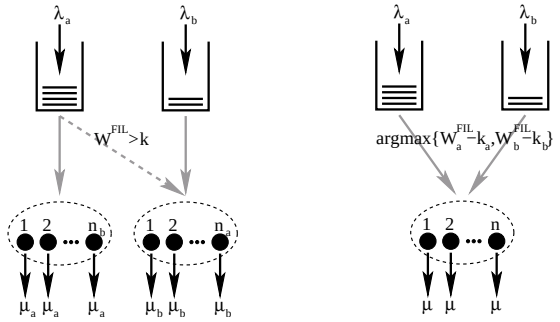
The traditional approach of modelling the number of customers in a system is not useful for modelling these systems as no information about the FIL waiting time is inherent in the variables. Instead we introduce the Erlang approximation (TEA) as a way of modelling the waiting time of the customer first in line. We show how TEA can be used for finding the waiting time distribution for different queueing systems.

The literature on this kind of prioritization is very limited despite its widespread use in the industry. In [2] a system is considered, where a single queue is served by two servers of which one are only allowed to take in customers when they have waited a given amount of time, and an expression for the waiting time distribution is found. In [1] a similar time-dependent overflow is approximated by a state-dependent overflow where states represent the number of customers in the system. Beside these, the literature is limited to less similar systems such as [14] where service times are exponentially distributed with parameters which depend on the waiting time experienced by the customer entering service.

In Section 2 TEA is presented for the simple case of an $M/M/n$ queue and it is shown how the waiting time distribution can be found. The principles behind TEA are easier to explain with this simple case and it also facilitates verification of the approximation by theoretical values.

In Section 3 it is shown how TEA can be used for modelling the N-design system with deterministic thresholds as shown in Figure 1a. The N-design is composed of a group of flexible servers and a group of specialized servers and two classes of customers. There exist multiple papers about the N-design; e.g. in [17] performance measures, such as average queueing delay and queue length distribution, are derived for a bilingual call center where a fraction of the agents are bilingual and the rest unilingual. In [16] the N-design is also examined and it is shown that a few flexible servers can improve performance significantly, when this is measured as the mean service time, however neither of these papers deal with the deterministic overflow threshold.

Further possibilities of TEA are discussed in Section 4, including an example of how the approach can be used for a system with low and high priority customers and scheduling according to due dates, also referred to as dynamic priority [11], [5]. Scheduling according to due date is a generalization of strict non-preemptive prioritization where high priority customers are not always given service first. This prioritization scheme is for example used for operating rooms scheduling to ensure lower priority operations will not experience exceedingly long waiting times while still keeping high priority patients' waiting times relatively low [3].



(a) N-design (b) V-design

Figure 1: Systems analyzed in Section 3 (Fig. 1a) and Section 4.1 (Fig. 1b).

Possible extensions of TEA are also discussed in Section 4. Only non-preemptive cases are considered here although the presented method could also be used for preemptive

(non-resume) prioritization. Finally a conclusion is given in Section 5.

2 The Erlang approximation

The fundamental idea of the Erlang approximation is to describe the FIL waiting time in a discretized form as states in a continuous time Markov chain (CTMC). The transition rate diagram of the CTMC for TEA of an $M/M/n$ queue is shown in Figure 2.

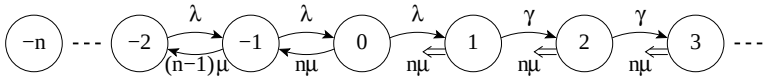


Figure 2: Transition rate diagram for the Erlang approximation of the $M/M/n$ system.

We define W_t^{FIL} as the waiting time of the first customer in line at time t, with the convention $W_t^{\text{FIL}} = 0$ when the queue is empty. In TEA we let states numbered $\{1, 2, ...\}$ represent $W_t^{\text{FIL}} > 0$. Transitions from state i to $i+1$ with rate γ represent the, in principle linearly, increasing FIL waiting time in a discrete way.

Whenever the queue is empty we have to differentiate between the number of free servers. This is done by letting states with negative numbers represent the number of free servers, thus state $-n$ corresponds to all n servers being vacant and state 0 to all servers being busy and the queue being empty. Arrivals occur with rate λ and are only "seen" by the model when the queue is empty. In the negative states, $i < 0$, where servers are unoccupied an arrival just goes to a free server and i increases by one. In state $i = 0$, were all servers are busy and the queue empty, an arriving customer enters the queue and becomes the first customer in line. This means W_t^{FIL} starts to increase, represented by the γ-transitions. Customers arriving when $i > 0$ are not seen by the model as these arrivals do not affect W_t^{FIL}.

Service completions occur with rate $n\mu$ for $i \geq 0$ and rate $(n+i)\mu$ for $i < 0$. Whenever a service completion occurs at time t and the queue is not empty, the FIL waiting decreases with $min\{S, W_t^{\text{FIL}}\}$, where S is a random variable describing the inter-arrival time of customers. In the considered $M/M/n$ queue, S is exponentially distributed as the inter-arrival times in a Poisson arrival process are exponential. This translates to a set of transitions from state i_{t_-} to states $\boldsymbol{i_{t_+}}$, following a geometric distribution in the discrete setting of TEA as given in Theorem 2.1. Here t_- and t_+ refer to the time just before and just after the transition at time t and $\boldsymbol{i_{t_+}} \in \{0 \ ... \ i_{t_-}\}$ is the set of states from 0 to i_{t_-}. These transitions are illustrated with the double arrows in Figure 2.

Theorem 2.1 *The discretization of the interarrival distribution is given as:*

$$P(I_{t_+} = i_{t_+} | I_{t_-} = i_{t_-}) = \begin{cases} 1 - \sum_{h=0}^{i_{t_-}-1} \left(\frac{\lambda}{\lambda+\gamma}\right)\left(\frac{\gamma}{\lambda+\gamma}\right)^h, & \text{for } i_{t_+} = 0; \\ \left(\frac{\lambda}{\lambda+\gamma}\right)\left(\frac{\gamma}{\lambda+\gamma}\right)^{i_{t_-}-i_{t_+}}, & \text{for } i_{t_+} \in \{1, .., i_{t_-}\}. \end{cases}$$

Proof The theorem follows directly from the geometric distribution, where the probability of having a γ-transition in a given state is $\gamma/(\lambda+\gamma)$. This means that the probability of going from state i_{t_-} to i_{t_+} equals the probability of having had $i_{t_-} - i_{t_+}$, γ-transitions

since the last λ-transition. The probability of a transition to state 0 is thus 1 minus the sum of probabilities of transitions to the other states. □

In order to determine the state probabilities at stationarity numerically, a finite state space is defined. This is done by truncating the number of phases describing the waiting time at state D, thus $i \in \{-n, ..., D\}$. By solving $\boldsymbol{\pi}\mathbf{G} = \mathbf{0}$, where $\mathbf{G}$ is a generator matrix containing the transition rates, the steady-state probabilities are given in the row vector $\boldsymbol{\pi}$. An example of the generator matrix is shown in (1) for an $M/M/2$ system with $D = 3$.

$$\mathbf{G} = \begin{bmatrix} -\sum & \lambda & 0 & 0 & 0 & 0 \\ \mu & -\sum & \lambda & 0 & 0 & 0 \\ 0 & 2\mu & -\sum & \lambda & 0 & 0 \\ 0 & 0 & p(0|1)2\mu & -\sum & \gamma & 0 \\ 0 & 0 & p(0|2)2\mu & p(1|2)2\mu & -\sum & \gamma \\ 0 & 0 & p(0|3)2\mu & p(1|3)2\mu & p(2|3)2\mu & -\sum \end{bmatrix}, \tag{1}$$

where $\sum$ represents the sums of the individual rows and the $p(j|i)'s$ are given in Theorem 2.1.

The truncation of the state space introduces the risk of having a large probability mass in the truncated state, especially if γ is large. The value of γ has a significant influence on the approximation. Increasing it means that more states are required for the truncation of states not to have a too significant influence on the precision of the approximation. However, having γ large at the same time improves the approximation as it then better represents the continuously elapsing time. This suggests having a threshold on probability mass in the truncated state, e.g. $\pi_D < 0.001$, otherwise γ as large as possible. The structure of $\mathbf{G}$ further suggests that the value of γ should be increasing with μ, n, D.

2.1 Waiting time distribution

In order to determine the waiting time distribution, the embedded Markov chain at service initiations is considered. Service initiations occur at λ-transitions from states with vacant servers, i.e. $i < 0$ and μ-transitions from states $i > 0$.

The state probability just before a service initiation from state i is denoted $\alpha(i)$. We let the distribution of $\alpha(i)$ be given in the vector $\boldsymbol{\alpha}$ which can be found as:

$$\alpha_{\mu\lambda}(i) = \frac{\pi(i)\Lambda_{\mu\lambda}(i)}{\sum_{j=-n}^{D} \pi(j)\Lambda_{\mu\lambda}(j)}, \tag{2}$$

Here $\Lambda(i)_{\mu\lambda}$ is the sum of the transition intensities from state (i) that result in a service initiation and the subscripts show the transitions that are considered. For the $M/M/n$ system $\Lambda(i)_{\mu\lambda}$ becomes

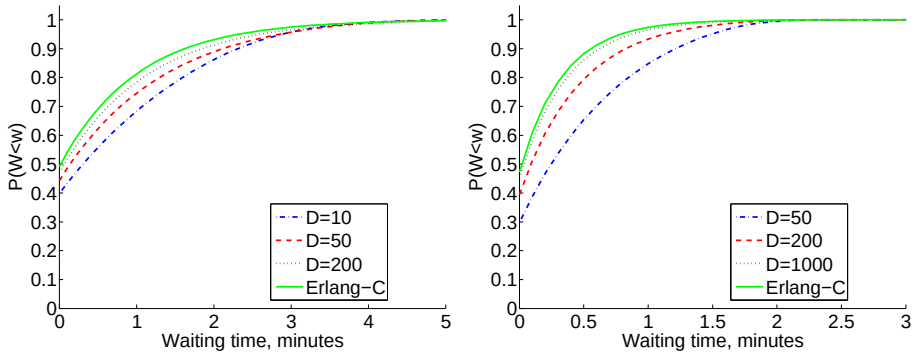
$$\Lambda_{\mu\lambda}(i) = \begin{cases} \lambda & \text{for } i < 0; \\ 0 & \text{for } i = 0; \\ n\mu & \text{for } 0 < i \leq D, \end{cases} \tag{3}$$

where we should note that $n\mu = \sum_{j=0}^{i} p(j|i)n\mu$. The waiting time distribution can now be found from (2) and (3). A customer entering service when $i < 0$ goes directly to

a free server and experiences no waiting time, this is represented by the first sum in Equation (4). When a customer enters service from a state $i > 0$, he/she has waited a sum of i exponentially distributed time periods, each with mean $1/\gamma$. Let $F_\Gamma(t; i, \gamma) = 1 - \sum_{h=0}^{i-1} \frac{(\gamma t)^h}{h!} e^{-\gamma t}$ be the cdf of an Erlang-distribution with shape parameter $i \in \mathbb{N}$ and scale parameter $\gamma \in \mathbb{R}_+$, then we have the second sum in Equation (4). The waiting time distribution, as approximated by TEA, of a customer entering the system then becomes:

$$P(W \leq t) = \sum_{i=-n}^{-1} \alpha_{\mu\lambda}(i) + \sum_{i=1}^{D} F_\Gamma(t; i, \gamma) \alpha_{\mu\lambda}(i), \tag{4}$$

Figure 3 shows the waiting time distribution as found by TEA for a small (3a) and a large system (3b) respectively, both compared to the theoretical distribution given as $F(t) = 1 - E_{2,n}(A)e^{-(n-A)\mu t}$, $A < n$, $t \geq 0$, where $A = \lambda/\mu$ is the offered traffic and $E_{2,n}(A)$ is the delay probability as given by Erlang's C-formula [8]. It can be seen that the approximation does indeed converge for $D \to \infty$, as desired.



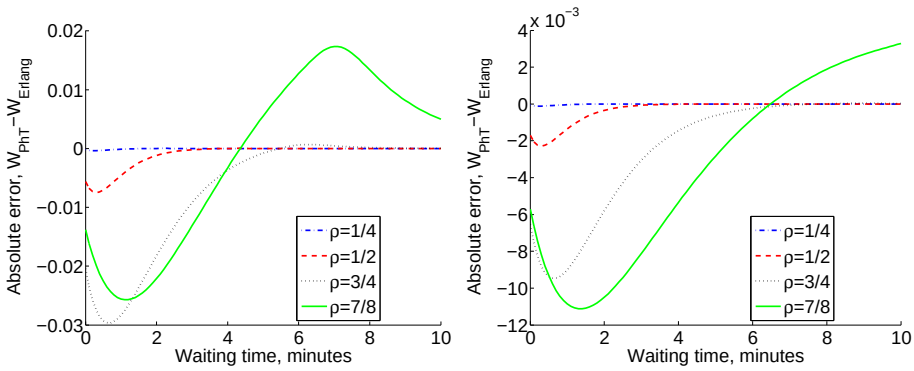
(a) $n = 4$, $\lambda = 3$, $\mu = 1$, $\gamma = \frac{2\mu}{3\lambda}(Dn)^{\frac{3}{4}}$ (b) $n = 40$, $\lambda = 37$, $\mu = 1$, $\gamma = \frac{3\mu n}{4\lambda} D^{0.9}$

Figure 3: The waiting time distribution as found by TEA compared to Erlang-C for a small (3a) and large system (3b) respectively.

In Figure 4 the absolute error for different values of the load on the system, ρ, is shown for two different model sizes, $D = 200$ (4a) and $D = 1000$ (4b). It is seen that TEA is able to handle different values of the load, ρ, well.

2.2 Infinite state space

Instead of truncating the state space, we can solve TEA of the $M/M/1$ queue for an infinite number of states. The generator matrix (1) bears a strong resemblance to the ditto for an $M/G/1$ system for $i \geq 1$, see [7], if we extend it to an infinite number of states. This motivates guessing a solution to the steady state probabilities in the form of $\pi(i) = \theta\beta^i(1 - \beta)$, where θ is a normalizing constant. Solving the infinite set of equations



(a) $D = 200$, $n = 4$, $\mu = 1$, $\gamma = \frac{2\mu}{3\lambda}(Dn)^{\frac{3}{4}}$ (b) $D = 1000$, $n = 4$, $\mu = 1$, $\gamma = \frac{2\mu}{3\lambda}(Dn)^{\frac{3}{4}}$

Figure 4: The absolute error of the waiting time distribution as found by TEA is compared to the corresponding distribution from Erlang-C for different loads. Fig. 4a shows a coarser approximation than Fig. 4b. Note that as, without loss of generality, $\mu = 1$ we have the load, ρ, equals λ/n.

$\boldsymbol{\pi}\mathbf{G} = \mathbf{0}$ yields an expression for the steady state probabilities;

$$\pi_i = \begin{cases} \dfrac{\mu\gamma(\mu-\lambda)}{\lambda^3+\mu^2\gamma} & , \text{ for } i = -1 \\ \dfrac{\lambda\gamma(\mu-\lambda)}{\lambda^3+\mu^2\gamma} & , \text{ for } i = 0 \\ \dfrac{\lambda^2(\mu-\lambda)}{\lambda^3+\mu^2\gamma}\left(\dfrac{\lambda+\gamma}{\mu+\gamma}\right)^i & , \text{ for } i \geq 1 \end{cases} \tag{5}$$

We can find the waiting time distribution from the steady state probabilities (5) in the same way as for the truncated system described in Section 2.1, the resulting expression becomes

$$P(W \leq t) = 1 - \frac{\lambda(\lambda+\gamma)}{\lambda^2+\mu\gamma} e^{\frac{\gamma}{\mu+\gamma}(\lambda-\mu)t}.$$

This turns out to converge to the known expression for the $M/M/1$ queue [8] for $\gamma \to \infty$, which proves that TEA does indeed converge to the correct solution.

3 Call center modelling

In this section it is shown how the Erlang approach introduced in Section 2 can be used for modelling the system shown in Figure 1a. This system is referred to as an N-design due to the outline of the routing scheme [6]. The system is motivated by the call handling often seen in call centers, where the servers would be agents answering calls from customers with different needs or importance.

In the setup, two job types, a and b, arrive at queue a and queue b with rates λ_a and λ_b respectively. Queue a is served by a group of n_a servers each handling jobs with

service rate μ_a; queue b is similarly served by n_b servers each working with service rate μ_b. When the waiting time, $W_{t,a}$, of the first job in queue a exceeds a limit, k, the job is allowed to go to server group b with non-preemptive priority over jobs in queue b. This means that there may be vacant b-servers even if there are a-customers waiting. As before γ-transitions represent the increasing FIL-waiting time with the addition that we let both FIL-waiting times increase simultaneously when jobs are queued in both queues.

The Erlang model used to describe this system is depicted in Figure 5. In this case the Erlang approximation presented in Section 2, is extended to two dimensions, one for each queue and server group. States are denoted (i, j), where i and j represent the a and b jobs respectively in the same way as for the $M/M/n$ case described in Section 2. We let $\mathcal{X}$ denote the complete set of states. The deterministic threshold, k, on the overflow is modelled by m phases after which the a-calls are allowed to go to the b-server group. In this 2-dimensional model the FIL-waiting times for queues a and b are truncated at states D_a and D_b respectively. It would be appealing to solve this system for an infinite number of states as was done for the $M/M/1$ queue in Section 2.2. However, the more complex generator matrix and the fact that one will have to deal with an infinite number of states in two dimensions would lead to complex analytical expressions, furthermore the additional insight gained would be limited.

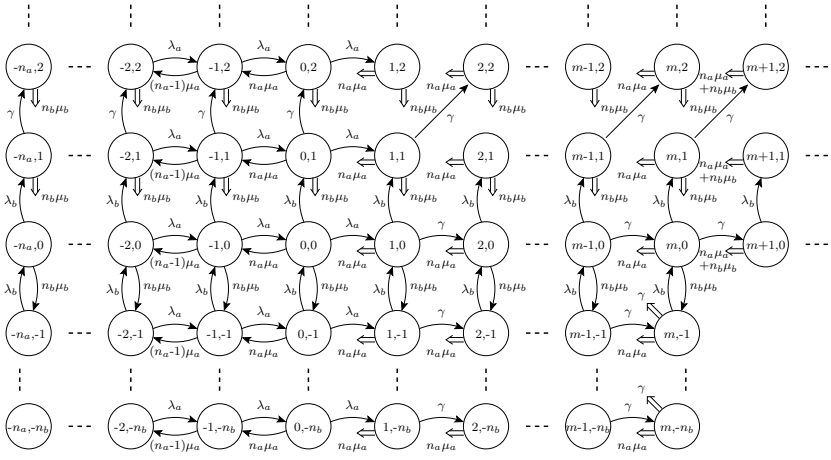


Figure 5: Illustration of the Erlang model. The horizontal and vertical directions describe queues a and b respectively. Negative indices refer to the number of vacant servers in each server group and positive indices refer to the waiting time of the first customer in line. The double arrows illustrate a distribution of transition intensities to a number of states in the given direction with total intensity as given next to the arrow. This distribution is given in Theorem 2.1.

When at least one of the servers in group b is vacant and queue a reaches state m, the next γ-transition will result in the first job in queue a being allowed to go to a free server in group b thus decreasing the waiting time of the first in line in queue a and decreasing the number of vacant servers in group b by one. This situation corresponds to the diagonal double arrows originating from states $(m, j < 0)$.

The use of Erlang distributions for approximating step functions in stochastic models often suffers from the curse of dimensionality. This is to some extent avoided by combining the number of free servers and the waiting time of the first in line in one dimension for each of the job types.

Extra care has to be taken when determining the waiting time distribution for the two job types. The transitions leading to service initiations in the N-design model are λ_a and λ_b-transitions from states with vacant servers, i.e. $i < 0$ and $j < 0$ respectively, γ-transitions from states where $i = m$ and $j < 0$, and finally μ_a and μ_b-transitions from states $i > 0$ and $j > 0$ respectively.

Besides the obvious differentiation between service initiations for a and b jobs, it is also necessary to deal with service initiations for a jobs due to γ transitions in a special way. This is due to γ-transitions representing customers having waited exactly k and not a gamma-distributed amount of time.

The state probabilities just before service initiations are denoted $\alpha(i,j)^a$ and $\alpha(i,j)^b$ for a and b-calls respectively. The distribution of $\boldsymbol{\alpha}$ can be found as:

$$\alpha^a_{\mu\lambda}(i,j) = \frac{\pi(i,j)\Lambda^a_{\mu\lambda}(i,j)}{\sum_{(i,j)\in\mathcal{X}} \pi(i,j)\Lambda^a_{\mu\lambda\gamma}(i,j)}, \tag{6}$$

$$\alpha^a_{\gamma}(m,j) = \frac{\pi(m,j)\Lambda^a_{\gamma}(m,j)}{\sum_{(i,j)\in\mathcal{X}} \pi(i,j)\Lambda^a_{\mu\lambda\gamma}(i,j)}, \qquad j < 0. \tag{7}$$

Here $\Lambda^a(i,j)$ is the sum of the transition intensities from state (i,j) that result in a service initiation for a-calls. The subscripts on Λ are used to differentiate between the different transitions that lead to service initiations. For a-calls, Λ is given as:

$$\Lambda^a_{\mu\lambda}(i,j) = \begin{cases} \lambda_a & \text{for } i < 0 \\ 0 & \text{for } i = 0 \\ n_a\mu_a & \text{for } 0 < i \leq m \\ n_a\mu_a + n_b\mu_b & \text{for } m < i \end{cases} \tag{8}$$

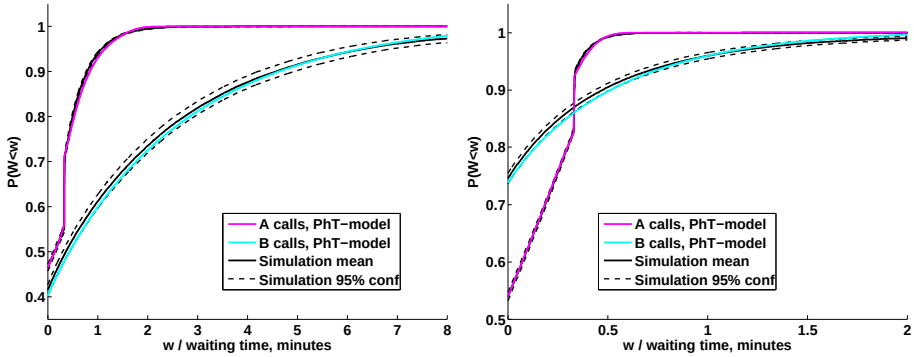
$$\Lambda^a_{\gamma}(i,j) = \begin{cases} \gamma & \text{for } i = m,\ j < 0; \\ 0 & \text{else.} \end{cases} \tag{9}$$

$$\Lambda^a_{\mu\lambda\gamma}(i,j) = \Lambda^a_{\mu\lambda}(i,j) + \Lambda^a_{\gamma}(i,j) \tag{10}$$

For the b-calls the expressions are somewhat simpler:

$$\Lambda^b_{\mu\lambda}(i,j) = \begin{cases} \lambda_b & \text{for } j < 0,\ i \leq m; \\ n_b\mu_b & \text{for } 0 < j,\ i \leq m \\ 0 & \text{else.} \end{cases} \tag{11}$$

Using the same approach as in Section 2, the waiting time distribution for type a jobs



(a) Small system (b) Large system

Figure 6: The Erlang approximation compared to simulations with 95% confidence intervals. The parameters used are $k = 0.33min.$, $\lambda_a = 0.75min^{-1}$, $\lambda_b = 1.75min^{-1}$, $\mu_a = 0.33min^{-1}$, $\mu_b = 0.5min^{-1}$, $\gamma = 30min^{-1}$, $n_a = 2$, $n_b = 5$, $m = 10$, $D_a = 50$ and $D_b = 300$ for Figure 6a and $k = 0.33min$, $\lambda_a = 7min^{-1}$, $\lambda_b = 10min^{-1}$, $\mu_a = 0.33min^{-1}$, $\mu_b = 0.5min^{-1}$, $\gamma = 90min^{-1}$, $n_a = 20$, $n_b = 27$, $m = 30$, $D_a = 40$ and $D_b = 200$ for Figure 6b.

can be found from (6) and (7) as:

$$P(W_a \leq t) = \begin{cases} \sum_{i=-n_a}^{-1} \sum_{j=-n_b}^{D_b} \alpha^a_{\mu\lambda}(i,j) + \sum_{i=1}^{D_a} \left[F_\Gamma(t;i,\gamma) \sum_{j=-n_b}^{D_b} \alpha^a_{\mu\lambda}(i,j) \right], & t < k, \\ \sum_{i=-n_a}^{-1} \sum_{j=-n_b}^{D_b} \alpha^a_{\mu\lambda}(i,j) + \sum_{j=-n_b}^{-1} \alpha^a_\gamma(m,j) & \\ \qquad + \sum_{i=1}^{D_a} \left[F_\Gamma(t;i,\gamma) \sum_{j=-n_b}^{D_b} \alpha^a_{\mu\lambda}(i,j) \right], & t \geq k, \end{cases}$$

The waiting time distribution for type b jobs can in a similar fashion be found as:

$$P(W_b \leq t) = \sum_{i=-n_a}^{D_a} \sum_{j=-n_b}^{-1} \alpha^b_{\mu\lambda}(i,j) + \sum_{j=1}^{D_b} \left[F_\Gamma(t;j,\gamma) \sum_{i=-n_a}^{D_a} \alpha^b_{\mu\lambda}(i,j) \right],$$

where the α^b can be found in the same way as for the a-jobs from Equation (6) using (11) instead of (8). γ-transitions never lead to service initiations for b-jobs thus leading to a simpler expression.

In Figure 6, TEA is compared to simulations for a small (Figure 6a) and a large system (Figure 6b). It is seen that TEA is able to give a good approximation even for large systems.

4 Further possibilities of TEA

The possibilities of TEA are not limited to the N-design presented in Section 3. In this section we show how a system with dynamic prioritization can be modelled using TEA and discuss how discretionary prioritization and the inclusion of abandonments in models could be approached. A setup with overflow to a back office, such as the one dealt with in [1], could also be analyzed with TEA.

4.1 Dynamic prioritization

In this section we show how TEA, introduced in Section 2, can be used for a system with dynamic priority. In this case the next customer to be served when a server becomes available at time t, is selected from two queues, a and b, by $max\{W_{t,a} - k_a, W_{t,b} - k_b\}$, where the k's are constants and the W_t's FIL-waiting times as previously. Arrivals are assumed to occur according to a Poisson process with rates λ_a and λ_b to the two queues. Jobs from both queues are served by a joint group of n servers that each handle the jobs with exponential service time with mean $1/\mu$. If one queue is empty, jobs from the other will always be taken into service immediately if a server is vacant. The system is illustrated in Figure 1b.

As is shown in [4], the average waiting time of the low priority customers in a static priority system with two classes grows quickly when the relative frequency of high-priority customers increases. This prioritization scheme is thus desirable when exceedingly long waiting times should be avoided for all classes while still giving some priority to certain customers.

The exact formulation of the additive prioritization scheme may vary as only the difference of the constants matters, adding k_a and k_b to $W_{t,b}$ and $W_{t,a}$ respectively would be equivalent. Here it is formulated by subtraction of a constant for each queue which especially for systems with more than two queues make things more clear as it can be interpreted as a target waiting time for each queue.

This prioritization scheme was first introduced in [9] and is referred to as dynamic priority or sequencing according to due-date in the literature. In [10], Jackson's conjecture is given, stating that the tails of the waiting time distributions for the different priority classes are exponential with the same shape. Indeed the shape of the tails are the same as for an FCFS system, only shifted $k_a - k_b$ from each other. The conjecture doesn't say anything about the remaining, lower part of the distribution, which is where TEA can give a good approximation.

The state space of TEA for the V-design system with dynamic priority becomes two-dimensional as the FIL-waiting times of both queues need to be taken into account. States $i < 0$ keep track of the number of vacant servers in the same way as for the $M/M/n$ system treated in Section 2. Whenever a server finishes a job and $i > 0$, the next customer taken into service must be chosen from $max\{W_{t,a} - k_a, W_{t,b} - k_b\}$. This is implemented in the model by dividing the state space into two parts representing either option as illustrated in Figure 7 with the thick dotted line.

In Figure 8, a plot of the waiting time distribution for a dynamic priority two class system is shown as found by TEA. Customers are always taken immediately into service when a server is vacant, thus the distributions start at the same value in 0. Figure 8 illustrates Jackson's conjecture [10] as it plots the waiting time distributions on a logarithmic scale

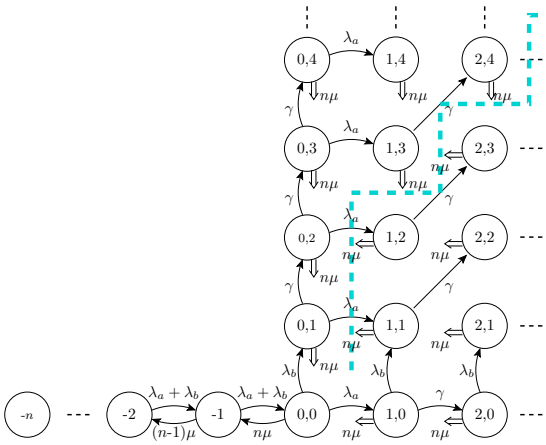


Figure 7: Transition rate diagram for modelling dynamic prioritization using TEA. The dotted line divides the state space into sections determining from which queue the next customer will be taken.

thus producing straight lines for the exponential tails. Simulation results with confidence intervals are shown for validation.

A possible extension to the dynamic priority model would be to keep servers free to cater for potential a-customers even though b-customers may be in line. This could be treated as an optimization problem and approached using Markov Decision Processes. It would be somewhat analogous to the slow-server problem [12], [13] and [15].

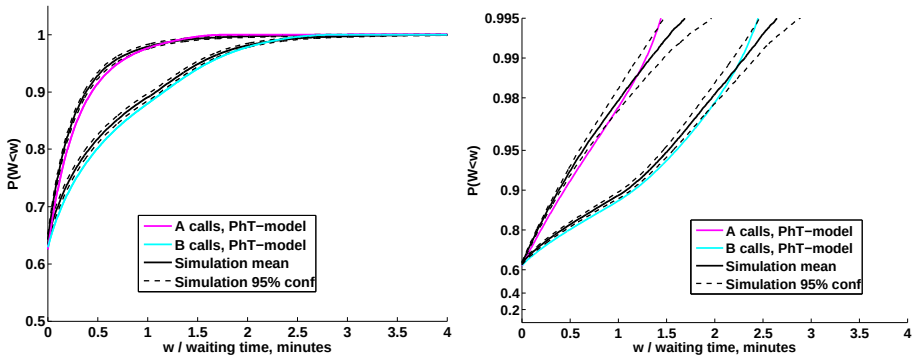
4.2 Discretionary priority

The discretionary priority discipline is a mix of the preemptive and non-preemptive disciplines, [11]. The idea is that high priority customers are only allowed to preempt lower priority customers under certain circumstances, e.g. that a low priority customer has just started service. In this case TEA could be used in an inverted way by modelling the elapsed service time of the last low priority customer who entered service and thus only allowing high priority customers to preempt when this value is under a given threshold.

4.3 Abandonments

The inclusion of abandonments is often considered essential when dealing with call centers [18]. It should be possible to extend the model to take abandonments into account by introducing transitions representing abandonments of the FIL customer. Also a negative binomial distribution should be used in Theorem 2.1 instead of the geometric distribution in order to account for the possibility of customers further back in the line having abandoned the system.

Including abandonments ought to improve the Erlang approximation as it will decrease the risk of the truncation having a significant negative influence.



(a) Waiting time distribution.

(b) Waiting time distribution, logarithmic plot.

Figure 8: The Erlang approximation compared to simulations for a system with dynamic priority. The parameters used are $k_a = 0.25$, $k_b = 1.25$, $\lambda_a = 3.5$, $\lambda_b = 2.5$, $\mu = 1$, $n = 8$, $D_a = 90$, $D_b = 150$, and $\gamma = 60$.

5 Conclusion

We introduced a new approach to modelling queueing systems by using the waiting time of the customer first in line. This has proved useful when dealing with systems where routing or priority of customers depends on this waiting time as seen in many real scenarios such as call centers.

The Erlang approximation (TEA) was introduced in Section 2 and we showed that the resulting waiting time distribution indeed converges to the theoretical value when the state space increases in size.

In Section 3 we implemented TEA for a N-design routing scheme with deterministic threshold on the overflow between server groups, a design often used in call centers. We showed that it is possible to get a good approximation of the waiting time distribution and that TEA can thus be an alternative approach to examining complicated systems where simulation studies have otherwise been seen as the only viable approach.

Further possibilities of TEA were discussed in Section 4 including how abandonments could be taken into account. We also showed how TEA can be used to model a system with dynamic priority, thus showing the flexibility of TEA.

References

[1] Wolfgang Barth, Michael Manitz, and Raik Stolletz. Analysis of Two-Level Support Systems with Time-Dependent Overflow - A Banking Application. Forthcoming in *Production and Operations Management*, 2009.

[2] René Bekker, Ger Koole, Bo Friis Nielsen, and Thomas Bang Nielsen. Queues with waiting time dependent service. Submitted for publication, June 2009.

[3] Margaret L. Brandeau, Franois Sainfort, and William P. Pierskalla. *Operations Research and Health Care: A Handbook of Methods and Applications*. Springer, illustrated edition, 2004.

[4] Alan Cobham. Priority assignment in waiting line problems. *Journal of the Operations Research Society of America*, 2(1):70–76, 1954.

[5] Richard W. Conway, William .L. Maxwell, and Louis W. Miller. *Theory of Scheduling*. Addison-Wesley Publishing Company, 1967.

[6] Noah Gans, Ger Koole, and Avishai Mandelbaum. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing and Service Operations Management*, 5(2):79–141, 2003.

[7] Geoffrey Grimmett and David Stirzaker. *Probability and Random Processes*. Oxford University Press, third edition, 2001.

[8] Villy Bæk Iversen. *Teletraffic Engineering and Network Planning*. COM Center, Technical University of Denmark, 2006.

[9] James R. Jackson. Some problems in queueing with dynamic priorities. *Nav. Res. Logistics Quart.*, 7:235–249, 1960.

[10] James R. Jackson. Queues with dynamic priority discipline. *Management Science*, 8(1):18–34 and 2627272, 1961.

[11] N.K. Jaiswal. *Priority Queues*. Academic Press, New York and Londons, 1968.

[12] G.M. Koole. A simple proof of the optimality of a threshold policy in a two-server queueing system. *Systems & Control Letters*, 26:301–303, 1995.

[13] W. Lin and P.R. Kumar. Optimal control of a queueing system with two heterogeneous servers. *IEEE Trans. Automat. Control*, 29:696–703, 1984.

[14] M.J.M Posner. Single-server queues with service time dependent on waiting time. *Operations Research*, 21:610–616, 1973.

[15] M. Rubinovitch. The slow server problem. *Journal of Applied Probability*, 22:205–213., 1985.

[16] Robert A. Shumsky. Approximation and analysis of a call center with flexible and specialized servers. *OR Spectrum*, 26(3):307–330, 2004.

[17] D.A. Stanford and W.K. Grassmann. The bilingual server system: a queueing model featuring fully and partially qualified servers. *INFOR*, 31(4):261–277, 1993.

[18] W. Whitt. Engineering solution of a basic call-center model. *Management Science*, 51(2):221–235, 2005.

B.1 Addendum

Verification of TEA

In this addendum to the paper *Waiting time dependent multi-server priority queues*, The Erlang Approximation is verified by using it with an infinite state space and finding the waiting time distribution of the $M/M/1$-queue. The transition rate diagram for this approximation is shown in Figure B.1

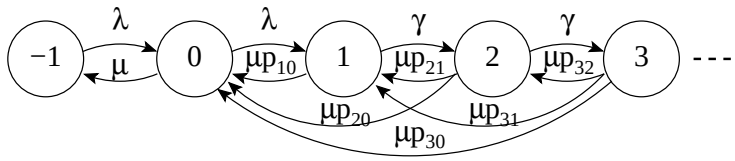


Figure B.1: Transition diagram of TEA applied to the $M/M/1$-queue.

The generator matrix for TEA applied to the $M/M/1$-queue is given by:

$$\mathbf{G} = \begin{bmatrix} -\lambda & \lambda & 0 & 0 & 0 & 0 & 0 \\ \mu & -(\lambda+\mu) & \lambda & 0 & 0 & 0 & 0 \\ 0 & \mu p_{10} & -(\gamma+\mu p_{10}) & \gamma & 0 & 0 & 0 \\ 0 & \mu p_{20} & \mu p_{21} & -(\gamma+\mu\sum_{i=0}^{1} p_{2i}) & \gamma & 0 & 0 \\ 0 & \mu p_{30} & \mu p_{31} & \mu p_{32} & -(\gamma+\mu\sum_{i=0}^{2} p_{3i}) & \gamma & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix} \tag{B.1}$$

The steady state probabilities can be found from solving the equation array $\mathbf{0} = \boldsymbol{\pi}\mathbf{G}$, where $\boldsymbol{\pi}$ is a row vector containing the steady state probabilities. The individual equations of this array can be written out as

$$\begin{aligned} 0 &= -\lambda\pi_{-1} + \mu\pi_0 \\ 0 &= \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \sum_{j=1}^{\infty} p_{0j}\mu\pi_j \\ 0 &= \lambda\pi_0 - (\gamma+\mu p_{01})\pi_1 + \sum_{j=2}^{\infty} p_{1j}\mu\pi_j \\ 0 &= \gamma\pi_{i-1} - \left(\gamma+\mu\sum_{j=0}^{i-1} p_{ij}\right)\pi_i + \sum_{j=i+1}^{\infty} p_{ij}\mu\pi_j, \quad \text{for } i \geq 2, \end{aligned}$$

where

$$p_{ji} = P(I_{t_+} = i | I_{t_-} = j) = \begin{cases} 1 - \sum_{h=0}^{j-1} \left(\frac{\lambda}{\lambda+\gamma}\right)\left(\frac{\gamma}{\lambda+\gamma}\right)^h, & \text{for } i = 0; \\ \left(\frac{\lambda}{\lambda+\gamma}\right)\left(\frac{\gamma}{\lambda+\gamma}\right)^{j-i}, & \text{for } i \in \{1, .., j\}, \end{cases} \tag{B.2}$$

are the probabilities describing the distribution of the transitions, whenever a customer goes into service.

Now, inserting (B.2) in each of the above equations yields the following equations:

$$0 = -\lambda\pi_{-1} + \mu\pi_0 \tag{B.3}$$

$$0 = \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \mu \sum_{j=1}^{\infty} \left(1 - \frac{\lambda}{\lambda+\gamma} \sum_{h=0}^{j-1} \left(\frac{\gamma}{\lambda+\gamma}\right)^h\right) \pi_j \tag{B.4}$$

$$0 = \lambda\pi_0 - \left(\gamma + \mu\left(1 - \frac{\lambda}{\lambda+\gamma}\right)\right)\pi_1 + \mu\left(\frac{\lambda}{\lambda+\gamma}\right) \sum_{j=2}^{\infty} \left(\frac{\gamma}{\lambda+\gamma}\right)^{j-1} \pi_j \tag{B.5}$$

$$\begin{aligned} 0 = \gamma\pi_{i-1} &- \left(\gamma + \mu\left(1 - \frac{\lambda}{\lambda+\gamma}\right)\right)\pi_i \\ &+ \mu\left(\frac{\lambda}{\lambda+\gamma}\right) \sum_{j=i+1}^{\infty} \left(\frac{\gamma}{\lambda+\gamma}\right)^{j-i} \pi_j, \quad \text{for } i \geq 2 \end{aligned} \tag{B.6}$$

The structure of $\mathbf{G}$, as seen in Equation (B.1), bears a strong resemblance to the generator matrix of an $M/G/1$-system [30]. This resemblance motivates guessing a solution of the form: $\pi_i = \theta\beta^i(1-\beta)$, for $i \geq 2$, i.e. a geometric distribution, where θ is a normalizing constant. Inserting this guess in Equation (B.6) enables us to find an expression for β:

$$
\begin{aligned}
0 &= \gamma\theta(1-\beta)\beta^{i-1} - \left(\gamma + \mu\left(1 - \frac{\lambda}{\lambda+\gamma}\right)\right)\theta(1-\beta)\beta^i \\
&\quad + \mu\left(\frac{\lambda}{\lambda+\gamma}\right)\sum_{j=i+1}^{\infty}\left(\frac{\gamma}{\lambda+\gamma}\right)^{j-i}\theta(1-\beta)\beta^j \\
0 &= \gamma\beta^{i-1} - \left(\gamma + \frac{\mu\gamma}{\lambda+\gamma}\right)\beta^i + \frac{\mu\lambda}{\lambda+\gamma}\sum_{j=i+1}^{\infty}\left(\frac{\gamma}{\lambda+\gamma}\right)^{j-i}\beta^j \\
0 &= \gamma - \left(\gamma + \frac{\mu\gamma}{\lambda+\gamma}\right)\beta + \frac{\mu\lambda\gamma}{(\lambda+\gamma)^2}\beta^2\frac{1}{1-\frac{\beta\gamma}{\lambda+\gamma}} \\
0 &= \gamma - \frac{\gamma^2}{\lambda+\gamma}\beta + \left(-\gamma - \frac{\mu\gamma}{\lambda+\gamma} + \frac{\beta\gamma^2}{\lambda+\gamma} + \frac{\mu\beta\gamma^2}{(\lambda+\gamma)^2}\right)\beta + \frac{\mu\lambda\gamma}{(\lambda+\gamma)^2}\beta^2 \\
0 &= \gamma - \left(\gamma + \frac{\mu\gamma+\gamma^2}{\lambda+\gamma}\right)\beta + \frac{\mu\lambda\gamma+\gamma^2(\lambda+\mu+\gamma)}{(\lambda+\gamma)^2}\beta^2 \\
0 &= \gamma(\lambda+\gamma)^2 - \left(\gamma(\lambda+\gamma)^2 + (\lambda+\gamma)(\mu\gamma+\gamma^2)\right)\beta + (\lambda+\gamma)(\mu\gamma+\gamma^2)\beta^2 \\
0 &= \lambda+\gamma - (\lambda+\mu+2\gamma)\beta + (\mu+\gamma)\beta^2,
\end{aligned}
$$

thus

$$
\beta = 1 \ \wedge \ \beta = \frac{\lambda+\gamma}{\mu+\gamma}. \tag{B.7}
$$

Now we consider Equation (B.5) and insert the guess $\pi_i = \theta(1-\beta)\beta^i$, for $i \geq 2$:

$$
\begin{aligned}
0 &= \lambda\pi_0 - \left(\gamma + \frac{\mu\gamma}{\lambda+\gamma}\right)\pi_1 + \mu\left(\frac{\lambda}{\lambda+\gamma}\right)\sum_{j=2}^{\infty}\left(\frac{\gamma}{\lambda+\gamma}\right)^{j-1}\theta(1-\beta)\beta^j \\
0 &= \lambda\pi_0 - \left(\gamma + \frac{\mu\gamma}{\lambda+\gamma}\right)\pi_1 + (1-\beta)\frac{\lambda\mu\gamma\theta\beta^2}{(\lambda+\gamma)^2}\frac{1}{1-\frac{\gamma\beta}{\lambda+\gamma}} \\
0 &= \left(\lambda - \frac{\lambda\gamma\beta}{\lambda+\gamma}\right)\pi_0 - \left(\gamma + \frac{\mu\gamma}{\lambda+\gamma} - \frac{\gamma^2\beta}{\lambda+\gamma} - \frac{\mu\gamma^2\beta}{(\lambda+\gamma)^2}\right)\pi_1 + (1-\beta)\frac{\lambda\mu\gamma\theta\beta^2}{(\lambda+\gamma)^2} \\
0 &= \left(\lambda(\lambda+\gamma)^2 - \lambda\gamma\beta(\lambda+\gamma)\right)\pi_0 \\
&\quad - \left(\gamma(\lambda+\gamma)^2 + \mu\gamma(\lambda+\gamma) - \gamma^2\beta(\lambda+\gamma) - \mu\gamma^2\beta\right)\pi_1 + (1-\beta)\lambda\mu\gamma\theta\beta^2 \\
0 &= \lambda(\lambda+\gamma)\left(\lambda+\gamma-\gamma\beta\right)\pi_0 - \gamma(\lambda+\mu+\gamma)(\lambda+\gamma-\gamma\beta)\pi_1 + \lambda\mu\gamma\theta(\beta^2-\beta^3).
\end{aligned} \tag{B.8}
$$

Inserting the non-trivial value of β from Equation (B.7) in Equation (B.8) yields

$$\begin{aligned}
0 &= \lambda(\lambda+\gamma)\left(\lambda+\gamma-\gamma\frac{\lambda+\gamma}{\mu+\gamma}\right)\pi_0 - \gamma(\lambda+\mu+\gamma)\left(\lambda+\gamma-\gamma\frac{\lambda+\gamma}{\mu+\gamma}\right)\pi_1 \\
&\quad + \lambda\mu\gamma\theta\left(\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^2 - \left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^3\right) \\
0 &= \lambda(\lambda+\gamma)\left(\frac{\mu}{\mu+\gamma}\right)\pi_0 - \gamma(\lambda+\mu+\gamma)\left(\frac{\mu}{\mu+\gamma}\right)\pi_1 \\
&\quad + \lambda\mu\gamma\theta\left(\frac{\lambda+\gamma}{(\mu+\gamma)^2} - \frac{(\lambda+\gamma)^2}{(\mu+\gamma)^3}\right) \\
0 &= \lambda(\lambda+\gamma)\pi_0 - \gamma(\lambda+\mu+\gamma)\pi_1 + \lambda\gamma\left(\frac{\lambda+\gamma}{\mu+\gamma} - \frac{(\lambda+\gamma)^2}{(\mu+\gamma)^2}\right)\theta \\
0 &= \frac{1}{\gamma}\pi_0 - \frac{\lambda+\mu+\gamma}{\lambda(\lambda+\gamma)}\pi_1 + \frac{\mu-\lambda}{(\mu+\gamma)^2}\theta \qquad \text{(B.9)}
\end{aligned}$$

By manipulating Equation (B.4) we get:

$$\begin{aligned}
0 &= \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \mu\sum_{j=1}^{\infty}\left(1 - \frac{\lambda}{\lambda+\gamma}\sum_{h=0}^{j-1}\left(\frac{\gamma}{\lambda+\gamma}\right)^h\right)\pi_j \\
0 &= \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \mu\sum_{j=1}^{\infty}\left(1 - \frac{\lambda}{\lambda+\gamma}\frac{1-\left(\frac{\gamma}{\lambda+\gamma}\right)^j}{1-\left(\frac{\gamma}{\lambda+\gamma}\right)}\right)\pi_j \\
0 &= \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \mu\sum_{j=1}^{\infty}\left(\frac{\gamma}{\lambda+\gamma}\right)^j\pi_j,
\end{aligned}$$

and inserting the guess for π_i yields

$$\begin{aligned}
0 &= \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \frac{\mu\gamma}{\lambda+\gamma}\pi_1 + \mu\sum_{j=2}^{\infty}\left(\frac{\gamma}{\lambda+\gamma}\right)^j\theta(1-\beta)\beta^j \\
0 &= \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \frac{\mu\gamma}{\lambda+\gamma}\pi_1 + \mu\theta(1-\beta)\left(\sum_{j=0}^{\infty}\left(\frac{\gamma\beta}{\lambda+\gamma}\right)^j - 1 - \frac{\gamma\beta}{\lambda+\gamma}\right) \\
0 &= \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \frac{\mu\gamma}{\lambda+\gamma}\pi_1 + \mu(1-\beta)\left(\frac{1}{1-\frac{\gamma\beta}{\lambda+\gamma}} - 1 - \frac{\gamma\beta}{\lambda+\gamma}\right)\theta.
\end{aligned}$$

The expression for β is inserted:

$$0 = \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \frac{\mu\gamma}{\lambda+\gamma}\pi_1 + \mu\left(1 - \frac{\lambda+\gamma}{\mu+\gamma}\right)\left(\frac{1}{1-\frac{\gamma\frac{\lambda+\gamma}{\mu+\gamma}}{\lambda+\gamma}} - 1 - \frac{\gamma\frac{\lambda+\gamma}{\mu+\gamma}}{\lambda+\gamma}\right)\theta$$

$$0 = \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \frac{\mu\gamma}{\lambda+\gamma}\pi_1 + \mu\left(\frac{\mu-\lambda}{\mu+\gamma}\right)\left(\frac{\mu+\gamma}{\mu} - \frac{\mu}{\mu+\gamma}\right)\theta$$

$$0 = \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \frac{\mu\gamma}{\lambda+\gamma}\pi_1 + \frac{\gamma(2\mu+\gamma)(\mu-\lambda)}{(\mu+\gamma)^2}\theta \tag{B.10}$$

We now consider Equations (B.3), (B.9), and (B.10) again:

$$0 = -\lambda\pi_{-1} + \mu\pi_0$$

$$0 = \frac{1}{\gamma}\pi_0 - \frac{\lambda+\mu+\gamma}{\lambda(\lambda+\gamma)}\pi_1 + \frac{\mu-\lambda}{(\mu+\gamma)^2}\theta$$

$$0 = \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \frac{\mu\gamma}{\lambda+\gamma}\pi_1 + \frac{\gamma(2\mu+\gamma)(\mu-\lambda)}{(\mu+\gamma)^2}\theta$$

From Equation (B.3), we express π_{-1} by π_0:

$$\pi_{-1} = \frac{\mu}{\lambda}\pi_0, \tag{B.11}$$

By inserting the expression of Equation (B.11) in Equation (B.10), we get:

$$0 = \lambda\pi_{-1} - (\lambda+\mu)\pi_0 + \frac{\mu\gamma}{\lambda+\gamma}\pi_1 + \frac{\gamma(2\mu+\gamma)(\mu-\lambda)}{(\mu+\gamma)^2}\theta$$

$$0 = \lambda\frac{\mu}{\lambda}\pi_0 - (\lambda+\mu)\pi_0 + \frac{\mu\gamma}{\lambda+\gamma}\pi_1 + \frac{\gamma(2\mu+\gamma)(\mu-\lambda)}{(\mu+\gamma)^2}\theta$$

$$0 = \lambda\pi_0 + \frac{\mu\gamma}{\lambda+\gamma}\pi_1 + \frac{\gamma(2\mu+\gamma)(\mu-\lambda)}{(\mu+\gamma)^2}\theta \tag{B.12}$$

Equation (B.9) is re-written as:

$$0 = \frac{1}{\gamma}\pi_0 - \frac{\lambda+\mu+\gamma}{\lambda(\lambda+\gamma)}\pi_1 + \frac{\mu-\lambda}{(\mu+\gamma)^2}\theta$$

$$\pi_0 = \frac{(\lambda+\mu+\gamma)\gamma}{\lambda(\lambda+\gamma)}\pi_1 - \frac{(\mu-\lambda)\gamma}{(\mu+\gamma)^2}\theta \tag{B.13}$$

Now, Equation (B.13) inserted in Equation (B.12) gives us an expression for π_1 by θ:

$$\begin{aligned}
0 &= \lambda\pi_0 + \frac{\mu\gamma}{\lambda+\gamma}\pi_1 + \frac{\gamma(2\mu+\gamma)(\mu-\lambda)}{(\mu+\gamma)^2}\theta \\
0 &= \lambda\left(\frac{(\lambda+\mu+\gamma)\gamma}{\lambda(\lambda+\gamma)}\pi_1 - \frac{(\mu-\lambda)\gamma}{(\mu+\gamma)^2}\theta\right) + \frac{\mu\gamma}{\lambda+\gamma}\pi_1 + \frac{\gamma(2\mu+\gamma)(\mu-\lambda)}{(\mu+\gamma)^2}\theta \\
0 &= \frac{(\lambda+2\mu+\gamma)\gamma}{\lambda+\gamma}\pi_1 - \frac{\gamma(\lambda+2\mu+\gamma)(\mu-\lambda)}{(\mu+\gamma)^2}\theta \\
0 &= \frac{1}{\lambda+\gamma}\pi_1 - \frac{\mu-\lambda}{(\mu+\gamma)^2}\theta \\
\pi_1 &= \frac{(\mu-\lambda)(\lambda+\gamma)}{(\mu+\gamma)^2}\theta
\end{aligned} \tag{B.14}$$

To obtain π_0 given by θ; Equation (B.14) is inserted in Equation (B.13):

$$\begin{aligned}
\pi_0 &= \frac{(\lambda+\mu+\gamma)\gamma}{\lambda(\lambda+\gamma)}\pi_1 - \frac{(\mu-\lambda)\gamma}{(\mu+\gamma)^2}\theta \\
\pi_0 &= \left(\frac{(\lambda+\mu+\gamma)\gamma}{\lambda(\lambda+\gamma)}\right)\left(\frac{(\mu-\lambda)(\lambda+\gamma)}{(\mu+\gamma)^2}\theta\right) - \frac{(\mu-\lambda)\gamma}{(\mu+\gamma)^2}\theta \\
\pi_0 &= \frac{(\mu+\gamma)\gamma(\mu-\lambda)}{\lambda(\mu+\gamma)^2}\theta \\
\pi_0 &= \frac{\gamma(\mu-\lambda)}{\lambda(\mu+\gamma)}\theta
\end{aligned} \tag{B.15}$$

Next, π_{-1} is found by inserting Equation (B.15) in Equation (B.11):

$$\begin{aligned}
\pi_{-1} &= \frac{\mu}{\lambda}\pi_0 \\
\pi_{-1} &= \frac{\mu\gamma(\mu-\lambda)}{\lambda^2(\mu+\gamma)}\theta
\end{aligned} \tag{B.16}$$

As the total probability mass of the states in equilibrium must sum to one, we

get:

$$
\begin{aligned}
1 &= \sum_{i=-1}^{\infty} \pi_i \\
1 &= \left[\frac{\mu\gamma(\mu-\lambda)}{\lambda^2(\mu+\gamma)} + \frac{\gamma(\mu-\lambda)}{\lambda(\mu+\gamma)} + \frac{(\mu-\lambda)(\lambda+\gamma)}{(\mu+\gamma)^2} + \sum_{i=2}^{\infty} \left(1 - \frac{\lambda+\gamma}{\mu+\gamma}\right) \left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i \right] \theta \\
1 &= \left[\frac{\mu\gamma(\mu-\lambda)}{\lambda^2(\mu+\gamma)} + \frac{\gamma(\mu-\lambda)}{\lambda(\mu+\gamma)} + \frac{(\mu-\lambda)(\lambda+\gamma)}{(\mu+\gamma)^2} + \frac{\lambda-\mu}{\mu+\gamma} \left(1 + \frac{\lambda+\gamma}{\mu+\gamma} - \frac{1}{1-\frac{\lambda+\gamma}{\mu+\gamma}}\right) \right] \theta \\
1 &= \left[\frac{\mu\gamma(\mu-\lambda)}{\lambda^2(\mu+\gamma)} + \frac{\gamma(\mu-\lambda)}{\lambda(\mu+\gamma)} + \frac{(\mu-\lambda)(\lambda+\gamma)}{(\mu+\gamma)^2} + \frac{(\lambda+\gamma)^2}{(\mu+\gamma)^2} \right] \theta \\
1 &= \frac{\mu\gamma(\mu-\lambda)(\mu+\gamma) + \gamma(\mu-\lambda)\lambda(\mu+\gamma) + \lambda^2(\mu-\lambda)(\lambda+\gamma) + \lambda^2(\lambda+\gamma)^2}{\lambda^2(\mu+\gamma)^2} \theta \\
1 &= \frac{(\lambda+\mu)\gamma(\mu-\lambda)(\mu+\gamma) + \lambda^2(\lambda+\gamma)(\mu+\gamma)}{\lambda^2(\mu+\gamma)^2} \theta \\
1 &= \frac{\lambda^3+\mu^2\gamma}{\lambda^2(\mu+\gamma)} \theta \\
\theta &= \frac{\lambda^2(\mu+\gamma)}{\lambda^3+\mu^2\gamma}
\end{aligned}
\tag{B.17}
$$

We can now find the explicit expressions for the steady state probabilities; first π_{-1}:

$$
\begin{aligned}
\pi_{-1} &= \frac{\mu\gamma(\mu-\lambda)}{\lambda^2(\mu+\gamma)} \theta \\
\pi_{-1} &= \left(\frac{\mu\gamma(\mu-\lambda)}{\lambda^2(\mu+\gamma)} \right) \left(\frac{\lambda^2(\mu+\gamma)}{\lambda^3+\mu^2\gamma} \right) \\
\pi_{-1} &= \frac{\mu\gamma(\mu-\lambda)}{\lambda^3+\mu^2\gamma},
\end{aligned}
\tag{B.18}
$$

then π_0:

$$
\begin{aligned}
\pi_0 &= \frac{\gamma(\mu-\lambda)}{\lambda(\mu+\gamma)} \theta \\
\pi_0 &= \left(\frac{\gamma(\mu-\lambda)}{\lambda(\mu+\gamma)} \right) \left(\frac{\lambda^2(\mu+\gamma)}{\lambda^3+\mu^2\gamma} \right) \\
\pi_0 &= \frac{\lambda\gamma(\mu-\lambda)}{\lambda^3+\mu^2\gamma},
\end{aligned}
\tag{B.19}
$$

and π_1:

$$\begin{aligned}\pi_1 &= \frac{(\mu-\lambda)(\lambda+\gamma)}{(\mu+\gamma)^2}\theta \\ \pi_1 &= \frac{(\mu-\lambda)(\lambda+\gamma)}{(\mu+\gamma)^2}\left(\frac{\lambda^2(\mu+\gamma)}{\lambda^3+\mu^2\gamma}\right) \\ \pi_1 &= \frac{\lambda^2(\mu-\lambda)(\lambda+\gamma)}{(\mu+\gamma)(\lambda^3+\mu^2\gamma)}. \end{aligned} \tag{B.20}$$

Finally π_i, for $i \geq 2$:

$$\begin{aligned}\pi_i &= \theta(1-\beta)\beta^i, && \text{for } i \geq 2 \\ \pi_i &= \frac{\lambda^2(\mu+\gamma)}{\lambda^3+\mu^2\gamma}\left(1-\frac{\lambda+\gamma}{\mu+\gamma}\right)\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i, && \text{for } i \geq 2 \\ \pi_i &= \frac{\lambda^2(\mu+\gamma)}{\lambda^3+\mu^2\gamma}\left(\frac{\mu-\lambda}{\mu+\gamma}\right)\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i, && \text{for } i \geq 2 \\ \pi_i &= \frac{\lambda^2(\mu-\lambda)}{\lambda^3+\mu^2\gamma}\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i, && \text{for } i \geq 2 \end{aligned}$$

We see that the expression for $i \geq 2$ is also valid for $i = 1$. To sum up, the steady state probabilities become:

$$\pi_i = \begin{cases} \dfrac{\mu\gamma(\mu-\lambda)}{\lambda^3+\mu^2\gamma} & \text{, for } i = -1 \\ \dfrac{\lambda\gamma(\mu-\lambda)}{\lambda^3+\mu^2\gamma} & \text{, for } i = 0 \\ \dfrac{\lambda^2(\mu-\lambda)}{\lambda^3+\mu^2\gamma}\left(\dfrac{\lambda+\gamma}{\mu+\gamma}\right)^i & \text{, for } i \geq 1 \end{cases}$$

With the state probabilities settled, we can now determine the waiting-time distribution. First we find the probabilities of having a service initiation from a state, i, given as α_i:

$$\alpha_i = \frac{\pi_i\Lambda_i}{\sum_{j=-1}^{\infty}\pi_j\Lambda_j}, \tag{B.21}$$

where

$$\Lambda_i = \begin{cases} \lambda & \text{for } i < 0; \\ 0 & \text{for } \ i = 0; \\ \mu & \text{for } \ i \geq 1, \end{cases}$$

is the transition intensity from each state that leads to a service initiation. The denominator in Equation (B.21) is common for all i, thus we start out with considering this:

$$\begin{aligned} \sum_{j=-1}^{\infty} \pi_j \Lambda_j &= \pi_{-1}\lambda + \sum_{j=1}^{\infty} \pi_j \mu \\ &= \frac{\mu\gamma(\mu-\lambda)}{\lambda^3+\mu^2\gamma}\lambda + \sum_{j=1}^{\infty} \frac{\lambda^2(\mu-\lambda)}{\lambda^3+\mu^2\gamma} \left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i \mu \\ &= \frac{\lambda\mu\gamma(\mu-\lambda)}{\lambda^3+\mu^2\gamma} + \frac{\lambda^2\mu(\mu-\lambda)}{\lambda^3+\mu^2\gamma}\frac{\lambda+\gamma}{\mu+\gamma}\frac{\mu+\gamma}{\mu-\lambda} \\ &= \frac{\lambda\mu\gamma(\mu-\lambda)}{\lambda^3+\mu^2\gamma} + \frac{\lambda^2\mu(\lambda+\gamma)}{\lambda^3+\mu^2\gamma} \\ &= \frac{\lambda^3\mu+\lambda\mu^2\gamma}{\lambda^3+\mu^2\gamma} \end{aligned}$$

From this, α_{-1} is determined to:

$$\begin{aligned} \alpha_{-1} &= \frac{\pi_i\Lambda_i}{\sum_{j=-1}^{\infty}\pi_j\Lambda_j}, \\ &= \frac{\mu\gamma(\mu-\lambda)}{\lambda^3+\mu^2\gamma}\lambda\frac{\lambda^3+\mu^2\gamma}{\lambda^3\mu+\lambda\mu^2\gamma} \\ &= \frac{\lambda\mu\gamma(\mu-\lambda)}{\lambda^3\mu+\lambda\mu^2\gamma} \\ &= \frac{\gamma(\mu-\lambda)}{\lambda^2+\mu\gamma} \end{aligned}$$

Obviously $\alpha_0 = 0$ as $\Lambda_0 = 0$. The α_i's, for $i \geq 1$, are found as:

$$\begin{aligned}
\alpha_i &= \frac{\pi_i \Lambda_i}{\sum_{j=-1}^{\infty} \pi_j \Lambda_j}, \\
&= \frac{\lambda^2(\mu-\lambda)}{\lambda^3+\mu^2\gamma} \left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i \mu \frac{\lambda^3+\mu^2\gamma}{\lambda^3\mu+\lambda\mu^2\gamma} \\
&= \frac{\lambda^2\mu(\mu-\lambda)}{\lambda^3\mu+\lambda\mu^2\gamma} \left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i \\
&= \frac{\lambda(\mu-\lambda)}{\lambda^2+\mu\gamma} \left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i
\end{aligned}$$

Now, we have all the elements needed to find the waiting-time distribution. It is given as:

$$P(W \leq t) = \alpha_{-1} + \sum_{i=1}^{\infty} F_\Gamma(t;,i,\gamma)\alpha_i,$$

where $F_\Gamma(t;i,\gamma) = 1 - \sum_{h=0}^{i-1} \frac{(\gamma t)^h}{h!} e^{-\gamma t}$ is the cdf of an Erlang-distribution with

shape parameter $i \in \mathbb{N}$ and scale parameter $\gamma \in \mathbb{R}_+$. Inserting the α's yield:

$$
\begin{aligned}
P(W \leq t) &= \alpha_{-1} + \sum_{i=1}^{\infty} F_\Gamma(t;,i,\gamma)\alpha_i \\
&= \frac{\gamma(\mu-\lambda)}{\lambda^2+\mu\gamma} + \sum_{i=1}^{\infty}\left(1 - \sum_{h=0}^{i-1}\frac{(\gamma t)^h}{h!}e^{-\gamma t}\right)\frac{\lambda(\mu-\lambda)}{\lambda^2+\mu\gamma}\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i \\
&= \frac{(\mu-\lambda)}{\lambda^2+\mu\gamma}\left[\gamma + \lambda\sum_{i=1}^{\infty}\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i - \lambda e^{-\gamma t}\sum_{i=1}^{\infty}\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i\sum_{h=0}^{i-1}\frac{(\gamma t)^h}{h!}\right] \\
&= \frac{(\mu-\lambda)}{\lambda^2+\mu\gamma}\left[\gamma + \lambda\frac{\lambda+\gamma}{\mu+\gamma}\left(\frac{\mu+\gamma}{\mu-\lambda}\right) - \lambda e^{-\gamma t}\sum_{i=1}^{\infty}\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i\sum_{h=0}^{i-1}\frac{(\gamma t)^h}{h!}\right] \\
&= 1 - \frac{\lambda(\mu-\lambda)}{\lambda^2+\mu\gamma}e^{-\gamma t}\sum_{i=1}^{\infty}\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i\sum_{h=0}^{i-1}\frac{(\gamma t)^h}{h!} \\
&= 1 - \frac{\lambda(\mu-\lambda)}{\lambda^2+\mu\gamma}e^{-\gamma t}\sum_{h=0}^{\infty}\frac{(\gamma t)^h}{h!}\sum_{i=h+1}^{\infty}\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i \\
&= 1 - \frac{\lambda(\mu-\lambda)}{\lambda^2+\mu\gamma}e^{-\gamma t}\sum_{h=0}^{\infty}\frac{(\gamma t)^h}{h!}\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^{h+1}\sum_{i=0}^{\infty}\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^i \\
&= 1 - \frac{\lambda(\mu-\lambda)}{\lambda^2+\mu\gamma}e^{-\gamma t}\sum_{h=0}^{\infty}\frac{(\gamma t)^h}{h!}\left(\frac{\lambda+\gamma}{\mu+\gamma}\right)^{h+1}\left(\frac{\mu+\gamma}{\mu-\lambda}\right) \\
&= 1 - \frac{\lambda(\lambda+\gamma)}{\lambda^2+\mu\gamma}e^{-\gamma t}e^{\frac{\gamma t(\lambda+\gamma)}{\mu+\gamma}} \\
&= 1 - \frac{\lambda^2/\gamma}{\lambda^2/\gamma+\mu} - \frac{\lambda}{\lambda^2/\gamma+\mu}e^{\frac{\gamma}{\mu+\gamma}(\lambda-\mu)t}
\end{aligned}
$$

which converges to $P(W \leq t) = 1 - \frac{\lambda}{\mu}e^{(\lambda-\mu)t}$ for $\gamma \to \infty$. The Erlang Approximation is verified by this result.

APPENDIX C

Optimization of overflow policies in call centers

Optimization of overflow policies in call centers

G.M. Koole[†], B.F. Nielsen[⋆], T.B. Nielsen[⋆]

[†]Dept. Mathematics
VU University Amsterdam
De Boelelaan 1081, 1081 HV, the Netherlands.

[⋆]Dept. Informatics and Mathematical Modelling
Technical University of Denmark
Richard Petersens Plads, 2800 Kgs. Lyngby, Denmark.

May 2, 2010

Abstract

We examine how overflow policies in a multi skill call center should be designed to accommodate different performance measures such Average Speed of Answer and Telephone Service Factor. This is done using a discrete Markovian approximation to the waiting time of the first customers waiting in line. We show that the widely used policy, of having a fixed threshold on the waiting time of the first customer in line determine when overflow is allowed, is far from optimal in many cases. Furthermore we discuss in which cases this approach is indeed appropriate. The work is motivated by call center applications but may be applicable to other fields such as health care management and communication networks.

Keywords: Waiting time distribution; call center; priority queues; Markov Decision Processes; optimization; Dynamic Programming.

1 Introduction

In call centers a commonly used practice is to allow overflow of calls between different agent pools. The purpose of this can be to reduce the risk of having calls waiting an excessive amount of time in one queue while servers assigned to another queue may be vacant. There can obviously be restrictions on the overflow between groups, e.g., due to one group of agents not having the competence to answer all types of calls or simply based on a wish of giving higher priority to one customer class.

The call routing structure of real call centers can often be very complicated with multiple customer classes and agent/server groups. In order to analyze these setups an often used practice is to look at different canonical structures, such as the "V", "N", "X", and "W" designs [4]. The system analyzed here is the one referred to as an N-design. Figure 1 shows an illustration of this system. Two customer classes, a and b, are each served by their own server group. Furthermore a-calls can be routed to b-servers according to decisions based on the number of free servers and/or the waiting times of the first customers in line in the two queues. b-calls can never go to a-servers. Also, a-calls cannot preempt b-calls already being served. It should be noted that a-calls are not necessarily of higher

priority than b-calls, the fundamental difference lies only in the routing scheme. For example, if a-calls are of high priority, the overflow to b-servers could be used to reduce the risk of having important customers wait too long in the queue. In the case where b-calls are of higher priority, the overflow from the a-queue could allow any excess capacity in the b-server group to be utilized. The prioritization of calls will be based on cost functions discussed later.

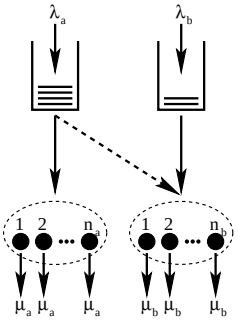


Figure 1: The N-design. a-customers can be sent to b-servers according to decisions based on the number of free servers and/or the waiting times of the first customers in line in the two queues.

An often used policy is to allow overflow after the waiting time of the first customer in line has reached a given fixed threshold, thus not taking into account if servers are vacant in the server group the overflow is directed to. This policy is examined in [3], [9] and [2]. The idea behind keeping b-servers vacant even though a-customers are waiting, is to have b-servers ready to answer incoming b-calls. However, the approach of letting the threshold where overflow is allowed to be independent of the number of free servers, is intuitively unattractive as it might very well be better to allow overflow earlier if many b-servers are vacant and later if few b-servers are vacant. In this paper we examine how overflow policies can be designed to accommodate different performance measures in the best way and thus avoid a waste of resources.

Different implementations of routing and call selection based on the waiting time of customers have been examined in the literature, especially for the V-design where two customer classes share a server group. One simple approach is to assign an ageing factor that grows proportionally with a customer's waiting time and choose the customer with the highest proportional waiting time. This approach was introduced in [8] and later also examined in [10]. A similar approach is to add a constant to the waiting time for each customer class and thus prioritize calls based on the values of the constant. This is often referred to as scheduling according to due dates or dynamic prioritization and was first examined in [6] and [7].

In [5], call selection is optimized and average costs are given for a single server and two customer classes with switching costs and preemptive-resume service discipline using one-step optimization. Non-preemptive systems with multiple customer classes are normally hard to analyze using Markov Decision Processes as knowledge of the customers currently being served is required. This requirement is relaxed here by modelling the waiting times of the first customers in line. As a result, the optimal policies will depend on the waiting time of the longest waiting call. This makes sense both from a theoretical and a practical

point of view, especially for objectives that are functions of the waiting time such as the percentage of calls that have waited shorter than a specific threshold (the Telephone Service Factor). One can imagine, and it is indeed shown in this paper, that a policy that uses actual waiting time information performs much better for this type of objective.

The model presented in this paper is also in other aspects more versatile than those examined in the literature. The cost functions can be chosen freely, and we consider the N-design instead of the more commonly examined V-design. The N-design can be seen as a more complicated version of call selection in a V-design system. When a b-server becomes vacant it has to decide whether to take an a- or b-customer into service, but with the complication of taking the number of working a-servers into account.

In Section 2 the model is introduced and the optimization process is described. Results for different cases are given in Section 3 and compared to the fixed threshold policy. Lastly in Section 4 we conclude on our work and give topics for further work.

2 Model description

We consider the system illustrated in Figure 1. Two customer classes, a and b, have their own queue and server group with n_a and n_b servers respectively. Overflow can be allowed from the a-queue to b-servers. Calls are taken from each of the two queues according to the First In, First Out (FIFO) principle. Arrivals to each of the queues happen according to Poisson processes with rates λ_a and λ_b. Exponential service times are assumed and individual servers in group a and b complete jobs with rates μ_a and μ_b, respectively. Service rates are thus associated with the servers, not the customers.

The underlying model we use is similar to the one used in [9], but with different overflow criteria. The system is modelled using a 2-dimensional continuous-time Markov chain where the two dimensions represent a- and b-calls, respectively. For a-calls we let states $i \leq 0$ represent the number of vacant a-servers, i.e., $i = -n_a$ corresponds to all a-servers being vacant and $i = 0$ corresponds to all being occupied. We use the second dimension to model b-calls according to the same principle.

When an arrival occurs and all servers are busy, the customer enters the queue and we start modelling the waiting time of the first customer in line. In this way arrivals are not explicitly handled by the model for states $i > 0$ as arrivals do not affect the waiting time of the first in line when customers are queued. States $i > 0$ thus represent the waiting time of the first customer in line, w^{FIL}, in a discrete manner. We let transitions with rate γ between state i and $i + 1$ represent elapsing time. Service completions are represented by a geometric distributed set of transitions from state i to states $\{0, 1, ..., i\}$ with total rate $n_a\mu_a$. The weights of the transitions are given in Lemma 2.1, where t_- and t_+ denote the instants just before and after a transition. Again b-calls are modelled in the same way.

Lemma 2.1

$$\mathbb{P}_a(I_{t_+} = i_{t_+} | I_{t_-} = i_{t_-}) = \begin{cases} 1 - \sum_{h=0}^{i_{t_-}-1} \left(\frac{\lambda_a}{\lambda_a+\gamma}\right)\left(\frac{\gamma}{\lambda_a+\gamma}\right)^h, & \text{for } i_{t_+} = 0; \\ \left(\frac{\lambda_a}{\lambda_a+\gamma}\right)\left(\frac{\gamma}{\lambda_a+\gamma}\right)^{i_{t_-}-i_{t_+}}, & \text{for } i_{t_+} \in \{1, .., i_{t_-}\}. \end{cases}$$

See [9] for proof of Lemma 2.1. We introduce the following notations: $p_{a,ij} = \mathbb{P}_a(I_{t_+} = j | I_{t_-} = i)$ and $p_{b,ij} = \mathbb{P}_b(J_{t_+} = j | J_{t_-} = i)$.

An approach for approximating the waiting time distribution for a system with a fixed overflow policy, is described in [9]. This approach can also be applied here. It involves the steady state distribution, $\boldsymbol{\pi}$, which is found by setting up a generator matrix, $\mathbf{G}$, according to the system description above, and solving the equation system $\boldsymbol{\pi}\mathbf{G} = \mathbf{0}$. Next, the state probabilities at the instants before service initiations, $\alpha(i,j)$, are found by considering the embedded Markov chain at service initiations. From this, the waiting time distribution for a-customers is found as

$$P(W_a \leq t) = \sum_{i=-n_b}^{D_b} \sum_{j=-n_a}^{-1} \alpha_a(i,j) + \sum_{i=1}^{D_a} \left[F_\Gamma(t;i,\gamma) \sum_{j=-n_b}^{D_b} \alpha_a(i,j) \right],$$

where $F_\Gamma(t;i,\gamma) = 1 - \sum_{h=0}^{i-1} \frac{(\gamma t)^h}{h!} e^{-\gamma t}$ is the cdf of an Erlang-distribution with shape parameter $i \in \mathbb{N}$ and scale parameter $\gamma \in \mathbb{R}_+$. The waiting time distribution for b-customers is found in a similar way.

2.1 Overflow optimization

By introducing decisions in the individual states, we can choose when to allow overflow from the a-queue to b-servers and thus optimize the system according to some chosen cost functions. For this purpose Markov Decision Processes are used. We let V_n be the total expected cost n steps from the horizon and use backwards recursion to determine the optimal policy [12], [11]. We base the costs on the waiting time experienced by customers, i.e., when a customer is taken into service after having waited a given time, this event is assigned a cost. As the waiting times of customers are represented by states with positive indices we introduce cost functions of the from $c_a(i)$ and $c_b(j)$. We also introduce the possibility of imposing a cost for allowing overflow, i.e., whenever an a-customer is sent to a b-server. This is denoted with c_s.

In the case where a-customers are queued and b-servers are free, i.e., states where $i > 0,\ j < 0$, we have the possibility of sending an a-customer to a b-server. We introduce decisions whenever there has been a transition to one of these states. These decisions are represented by the W's given in Equation (1). The first part of the minimization term corresponds to doing nothing and staying in the state in which the system just arrived. The second part corresponds to letting one of the vacant b-servers take an a-customer into service, which then happens immediately after the decision has been taken.

$$W_n(i,j) = \begin{cases} \min \left\{ V_{n-1}(i,j),\ c_s + c_a(i) + \sum_{h=0}^{i} p_{a,ih} V_{n-1}(h,j+1) \right\} & \text{for } i > 0, j < 0 \\ V_{n-1}(i,j) & \text{else.} \end{cases} \quad (1)$$

A slightly different approach to decisions is taken for the case where customers are queued in both queues, i.e., states $i > 0,\ j > 0$. Here all servers are working and decisions are only allowed whenever a b-server finishes a job, that is, when a μ_b-transition occurs. We introduce decisions represented by $U(i,j)$ when a b-server finishes a job in state (i,j). The option here is whether to take an a- or b-customer into service, as represented by U in Equation (2).

$$U_n(i,j) = \min \left\{ c_b(j) + \sum_{h=0}^{j} p_{b,jh} V_{n-1}(i,h),\ c_s + c_a(i) + \sum_{h=0}^{i} p_{a,ih} V_{n-1}(h,j) \right\} \quad (2)$$

For $i \leq 0$, we have no decisions to make as no a-calls are queued. For states $i > 0,\ j = 0$ there is also no decision as no b-servers are free and if one becomes free a transition is made to a state $j = -1$, which the next decision will be based on.

Having defined the decisions, we can now set up the equations for the backward recursion. Five different equations are needed due to the structure of the state space. These are given in Equations (3)-(7). In order to have the same average time between transitions from each individual state we introduce uniformization terms. These are given in the last line in each of the equations giving a uniformization rate of $\gamma + \lambda_a + \lambda_b + n_a\mu_a + n_b\mu_b$.

$$\begin{aligned} V_n(i,j) &= \lambda_a W_n(i+1,j) + \lambda_b V_{n-1}(i,j+1) \\ &\quad + \mu_a(n_a+i)V_{n-1}(i-1,j) + \mu_b(n_b+j)V_{n-1}(i,j-1) \\ &\quad + (-i\mu_a - j\mu_b + \gamma)V_{n-1}(i,j) \qquad \text{for } i \leq 0, j \leq 0 \end{aligned} \tag{3}$$

$$\begin{aligned} V_n(i,j) &= \gamma V_{n-1}(i,j+1) + \lambda_a V_{n-1}(i+1,j) \\ &\quad + \mu_a(n_a+i)V_{n-1}(i-1,j) + \mu_b n_b \left(c_b(j) + \sum_{h=0}^{j} p_{b,jh} V_{n-1}(i,h) \right) \\ &\quad + (\lambda_b - i\mu_a)V_{n-1}(i,j) \qquad \text{for } i \leq 0, j > 0 \end{aligned} \tag{4}$$

$$\begin{aligned} V_n(i,j) &= \gamma W_n(i+1,j) + \lambda_b W_n(i,j+1) \\ &\quad + n_a\mu_a \left(c_a(i) + \sum_{h=0}^{i} p_{a,ih} W_n(h,j) \right) + (n_b+j)\mu_b W_n(i,j-1) \\ &\quad + (\lambda_a - j\mu_b)V_{n-1}(i,j) \qquad \text{for } i > 0, j < 0 \end{aligned} \tag{5}$$

$$\begin{aligned} V_n(i,j) &= \gamma V_{n-1}(i+1,j) + \lambda_b V_{n-1}(i,j+1) \\ &\quad + n_a\mu_a \left(c_a(i) + \sum_{h=0}^{i} p_{a,ih} V_{n-1}(h,j) \right) + n_b\mu_b W_n(i,j-1) \\ &\quad + \lambda_a V_{n-1}(i,j) \qquad \text{for } i > 0, j = 0 \end{aligned} \tag{6}$$

$$\begin{aligned} V_n(i,j) &= \gamma V_{n-1}(i+1,j+1) \\ &\quad + n_a\mu_a \left(c_a(i) + \sum_{h=0}^{i} p_{a,ih} V_{n-1}(h,j) \right) + n_b\mu_b U_n(i,j) \\ &\quad + (\lambda_a + \lambda_b)V_{n-1}(i,j) \qquad \text{for } i > 0, j > 0 \end{aligned} \tag{7}$$

We let the value iteration algorithm run until $V_n - V_{n-1}$ converges as follows, where ϵ is some small number, e.g. 10^{-9}:

$$\max_{(i,j)} \{V_n(i,j) - V_{n-1}(i,j)\} - \min_{(i,j)} \{V_n(i,j) - V_{n-1}(i,j)\} < \epsilon.$$

When the optimal policy has been determined through value iteration, we end up with a set of decisions, one for each of the states $i > 0,\ j < 0$ and $i > 0,\ j > 0$. A generator matrix for the system can then be set up based on these decisions. This is required to determine the stationary distribution and from this, the waiting time distribution of customers as described earlier in this section. Performance measures such as TSF and ASA can readily be extracted from the waiting time distributions.

2.2 Performance measures

The prevalent way of measuring performance in call centers is to use Average Speed of Answer (ASA) and Telephone Service Factor (TSF), the latter being the percentage of calls whose waiting times fall below a service level target. The cost functions for a and b-calls can be designed to emulate either, or both of these. Furthermore it may be appropriate to penalize overflow as it might be desired that a-calls are assigned to a-servers and b-calls to b-servers. The actual form and value of the cost functions $c_a(i)$ and $c_b(j)$ together with the overflow/switching cost c_s will always be a managerial decision based on often subjective priorities.

One might choose $c_a(i)$ and $c_b(j)$ as step functions being 0 below a certain threshold and some non-zero value above this threshold to represent a given TSF such as 80/20 (percentage/service level target), where we introduce SLT_a and SLT_b as the service level targets for a- and b-customers respectively. Another possibility is to let the cost functions increase linearly, which would then correspond to having a performance measure based on ASA. A combination of the two performance measures is of course also possible. A call center manager might choose to value a-calls twice as much as b-calls and thus let the cost function for a-calls increase twice as fast. However, a-calls need not necessarily be the high priority customers as discussed earlier.

The optimization of the routing policy is done under a given set of parameters, thus our target is not to dimension the system to fulfill a given TSF but rather doing as good as possible under given conditions. That is, serving as many as possible within the time limit in the TSF of the two customer classes with the possibility of having more weight on serving one class. Or, in the case where the cost functions are shaped to emulate an emphasis on ASA, trying to keep this as low as possible.

The way to relate states in the model to actual waiting times is to use the mean time spent in a state before a γ-transition happens, i.e., $1/\gamma$. For example, if $\gamma = 30\ min^{-1}$ then being in state $i > 0$ will correspond to the first customer in line having waited approximately $2i$ seconds.

3 Results

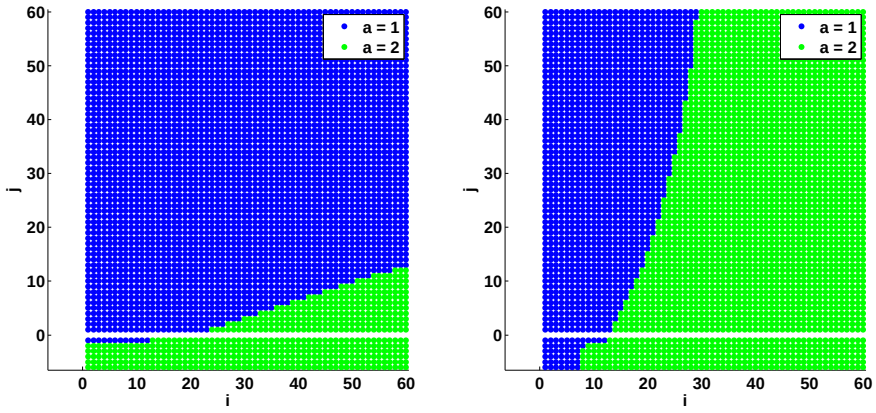
In this section we illustrate the optimal policy for a number of different cases. The policies are illustrated by plots of the state space with colors representing the optimal decision in each state. In all cases $a = 1$ (blue) indicates that no action should be taken and $a = 2$ (green) indicates that a call should be taken from the a-queue and assigned to a b-server. The white areas indicate that no decision is possible. The policy of allowing overflow after a given fixed time would in this case correspond to the state space being divided in a blue section to the left and green section to the right where the border would be a vertical line situated at the threshold value. No units are given on the parameters or results as these can be chosen as desired. However, the values have been chosen such that they can realistically be assumed to be in minutes (or min^{-1} for the intensities) in a call center context.

For all results, a larger than shown state space has been used in order to avoid the influence of the truncated state space. Typically a state space of size 120×120 has been used in the calculations to produce valid results for a state space of size 70×70. The results have been verified by setting $c_s = \infty$, which leads to overflow never being allowed. This corresponds to having two independent $M/M/n$ queues and the resulting waiting

time distribution from the model can thus be compared to analytical results. The model has also been compared to the simulations used for verification of the fixed threshold case in [9], and again the results agree.

Figure 2a shows an optimal policy when the cost functions are based on minimizing ASA and both types of calls are assigned the same weight or value. It is almost always optimal to route a-calls to b-servers when some of these are available, which makes sense as this minimizes the time servers are free while calls are waiting. When calls are waiting in both queues the optimal policy is in general to take the call that has waited the longest with a small bias towards b-customers. This is due to that fact that it is not desirable to end up in a states $i < 0$ where some servers will not be working.

Figure 2b shows what happens when a penalty is given when calls are allowed to overflow and the mean waiting time of a-calls are given triple weight compared to the mean for b-customers. It is seen that the decision to allow overflow is only taken when the waiting time of a-customers has reached a certain value. In this case the optimal policy actually bears some resemblance to the often used policy of allowing overflow after a fixed amount of time, with the difference that the border separating the areas with different decisions is not totally vertical.



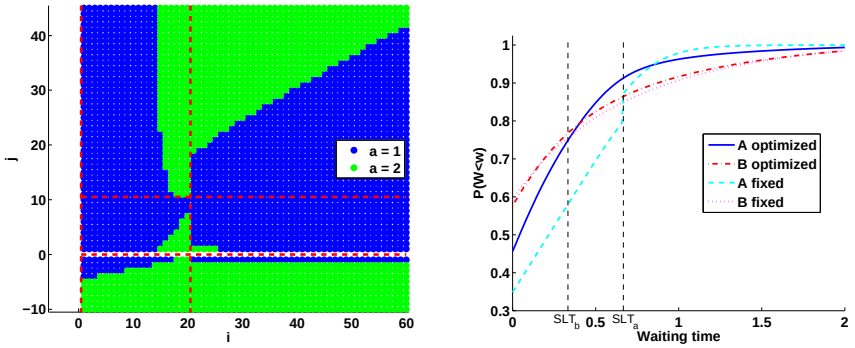
(a) Optimal policy, equal priority

(b) High priority a-calls, penalty to overflow

Figure 2: Decisions based on optimization of ASA. Green shows where overflow is allowed. Unique parameters for Figure 2a are: $c_a(i) = i,\ c_s = 0$, and for Figure 2b $c_a(i) = 3i,\ c_s = 120$. Shared parameters for the two figures are $c_b(j) = j,\ \lambda_a = 5;\ \lambda_b = 4;\ \mu_a = 1;\ \mu_b = 1;\ n_a = 6;\ n_b = 6;\ \gamma = 30;\ \epsilon = 10^{-9}$.

The optimal policy when using TSF as performance indicator can take a rather complicated form as seen in Figure 3a. If lines are overlaid at SLT_a, SLT_b, $i = 0$, and $j = 0$, as is done with red dotted lines in Figure 3a, the picture becomes more clear. Within each of these seven areas it can actually be seen that the decisions do indeed follow a monotone pattern.

The cost functions for Figure 3 have been chosen such that the b-customers are the ones requiring shorter waiting times (below $10s$) and a-customers are of lower priority and can wait up to $20s$ before a penalty is given. In this case it can clearly be seen that when a-customers have not waited for a long time, they should only be sent to b-servers if many of these are vacant. If customers are waiting in both queues, overflow should



(a) Optimal policy (b) Waiting time distributions

Figure 3: Optimal policy and waiting distributions when using TSF as performance indicator. The used parameters are: $c_a(i) = 0$ for $i \leq 20$, $c_a(i) = 1$ for $i > 20$, $c_b(j) = 0$ for $j \leq 10$, $c_b(j) = 1$ for $j > 10$, $c_s = 0$, $\lambda_a = 5$; $\lambda_b = 7$; $\mu_a = 1$; $\mu_b = 1$; $n_a = 5$; $n_b = 10$; $\gamma = 30$; $\epsilon = 10^{-9}$. The vertical lines in Figure b show the service level thresholds for a- and b-customers.

only be allowed if the first in line in the a-queue is just about to reach its service level threshold. In the case when both types of customers have exceeded their threshold, we have a diagonal division of the state space, roughly corresponding to taking the customer with the shortest waiting time of the two situated first in line into service. In this way, the probability that the next customer from that queue will be served within its threshold is maximized.

Figure 3b shows that the number of customers served within their threshold is increased for both a and b customers. The tail of the waiting time distribution for the optimized case of a-customers is longer than that of the fixed threshold case, but this does not matter when the focus is entirely on improving TSF. Also less b-customers are served immediately using the optimized policy, but again TSF is improved. Table 1 shows the actual numbers. ASA is also improved for both classes even though it is not taken into account during the optimization in this case.

Table 1: Numerical results; values from Figure 3.

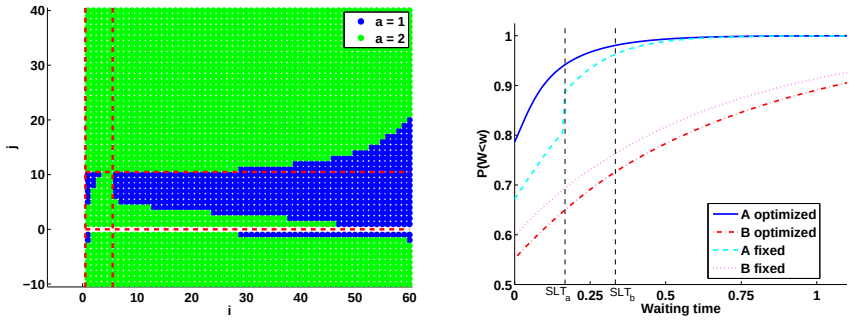
Policy	TSF_a	TSF_b	ASA_a	ASA_b
Fixed	87.25%	75.60%	0.3071	0.2274
Optimized	91.29%	76.79%	0.2074	0.2152

Figure 4 shows a case where a-customers are given high priority in the way that they are valued twice as much as b-customers. Furthermore the service level targets have been set to what would correspond to $10s$ and $20s$ for a- and b-customers respectively if we assume the parameters are taken to be in minutes. The results show that it is almost always advantageous to allow overflow when a-customers are given high priority. From the structure of the optimal policy shown in Figure 4a it appears that instead of having the waiting time the first in line of a-customers determine when overflow should be allowed, one should base the decision on the waiting time of the b-customers. Introducing a simple

policy where overflow is allowed when b-servers are available ($j < 0$) and when the waiting time of the first b-customer in line, w_b^{FIL}, has exceeded SLT_b, i.e., $w_b^{\text{FIL}} > SLT_b$, would resemble the picture seen in Figure 4a to a high degree. The optimal policy results in a significant improvement for a-customers at a minor cost to b-customers as seen in Figure 4b. The actual numbers are given in Table 2.

Table 2: Numerical results; values from Figure 4.

Policy	TSF_a	TSF_b	ASA_a	ASA_b
Fixed	88.76%	76.36%	0.0603	0.2151
Optimized	94.18%	72.63%	0.0297	0.2504



(a) Optimal policy

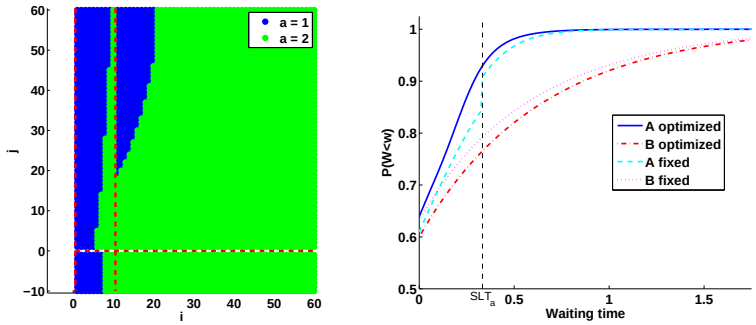
(b) Waiting time distributions

Figure 4: Optimal policy and waiting time distributions when using TSF as performance indicator and a-customers are given high priority. The used parameters are: $c_a(i) = 0$ for $i \leq 5$, $c_a(i) = 2$ for $i > 5$, $c_b(j) = 0$ for $j \leq 10$, $c_b(j) = 1$ for $j > 10$, $c_s = 0$, $\lambda_a = 5$; $\lambda_b = 7$; $\mu_a = 1$; $\mu_b = 1$; $n_a = 6$; $n_b = 10$; $\gamma = 30$; $\epsilon = 10^{-9}$. The vertical lines in Figure b show the service level thresholds for a- and b-customers.

A realistic cost function would be a combination of TSF and ASA as both are often used as performance measures in call centers. Figure 5 shows a case where the cost functions are based on a combination of the two with a penalty for overflow. Again the optimal policy has some resemblance to the fixed threshold, especially if the blue area for $i > 10$, $j > 0$ is disregarded. This indicates that the use of a fixed threshold policy corresponds to having high-priority a-customers and a penalty for overflow. We see that if a fixed threshold is used, it should be set to a lower value than SLT_a.

Table 3: Numerical results; values from Figure 5.

Policy	TSF_a	TSF_b	ASA_a	ASA_b
Fixed	90.90%	88.02%	0.1033	0.1919
Optimized	92.97%	86.23%	0.0809	0.2157



(a) Optimal policy

(b) Waiting time distributions

Figure 5: A combination of ASA and TSF as performance indicator is used here. The used parameters are: $c_a(i) = 3i$ for $i \leq 10$, $c_a(i) = 3i + 40$ for $i > 10$, $c_b(j) = j$; $c_s = 120$, $\lambda_a = 5$; $\lambda_b = 7$; $\mu_a = 1$; $\mu_b = 1$; $n_a = 6$; $n_b = 10$; $\gamma = 30$; $\epsilon = 10^{-9}$. The vertical line in Figure 3b shows the service level threshold for a-customers.

4 Conclusion

In this paper, we have presented a method for optimizing the overflow in queueing systems based on a call center scenario. We have shown that it is possible to obtain a significant performance improvement by using more complicated overflow policies compared to the simple fixed threshold.

The practice of not taking a-customers into service no matter how many free b-servers there are, is not recommendable. If a-customers are of high priority, it is almost always optimal to allow overflow to vacant b-servers, especially if no penalty is given to the overflow. However, in the case where b-customers are of higher priority it may indeed be better to reserve some servers for potential b-arrivals even if a-customers are waiting as illustrated in Figure 3. A general rule for when overflow should be allowed based on the number of free b-servers will most likely depend on many parameters and is a topic of further research.

The policy of using a fixed threshold for determining when to allow overflow has been shown to be a good approach when a penalty is associated with overflow and a-customers are of high priority. This result gives an idea of what the consequence is when the policy is used in call centers.

In all cases it appears that a fixed threshold should not be set to the same value as the service level threshold, but to a lower value. This makes sense, as some time is needed for b-servers to have a chance to become free before the threshold is reached, when calls are queued in both queues.

A natural and fairly straightforward extension of the model would be to allow overflow from the b-queue to a-servers and optimize both overflows simultaneously according to cost functions. This would constitute an X-design, which is one of the other canonical designs referred to in the introduction.

A final subject for further research is including fairness between servers in the objective function, i.e., trying to balance the occupation level of the servers. This is highly relevant for practical purposes, but it is hard to include in the dynamic programming formulation.

References

[1] Mor Armony and Nicholas Bambos. Queueing dynamics and maximal throughput scheduling in switched processing systems. *Queueing Systems*, 44(3):209–252, 2003.

[2] Wolfgang Barth, Michael Manitz, and Raik Stolletz. Analysis of two-level support systems with time-dependent overflow - a banking application. Forthcoming in *Production and Operations Management*, 2009, available from http://ideas.repec.org/p/han/dpaper/dp-399.html.

[3] René Bekker, Ger Koole, Bo Friis Nielsen, and Thomas Bang Nielsen. Queues with waiting time dependent service. Submitted for publication, 2009, available from http://www.math.vu.nl/ koole/research/.

[4] Noah Gans, Ger Koole, and Avishai Mandelbaum. Telephone call centers: Tutorial, review, and research prospects. *Manufacturing and Service Operations Management*, 5(2):79–141, 2003.

[5] Robin Groenevelt, Ger Koole, and Philippe Nain. On the bias vector of a two-class preemptive priority queue. *Mathematical Methods of Operations Research*, 55(1):107–120, 2002.

[6] James R. Jackson. Some problems in queueing with dynamic priorities. *Nav. Res. Logistics Quart.*, 7:235–249, 1960.

[7] James R. Jackson. Queues with dynamic priority discipline. *Management Science*, 8(1):18–34 and 2627272, 1961.

[8] Leonard Kleinrock and Roy P. Finkelstein. Time dependent priority queues. *Operations Research*, 15(1):104–116; 168514, 1967.

[9] Ger Koole, Bo Friis Nielsen, and Thomas Bang Nielsen. Waiting time dependent multi-server priority queues. Submitted for publication, 2009, available from http://www.math.vu.nl/ koole/research/.

[10] M. Perry and A. Nilsson. Performance modelling of automatic call distributors: assignable grade of service staffing. *International Switching Symposium 1992. 'Diversification and Integration of Networks and Switching Technologies Towards the 21st Century' Proceedings*, pages 294–8 vol.2, 1992.

[11] Martin L. Puterman. *Markov Decision Processes. Discrete Stochastic Dynamic Programming*. Wiley, second edition, 2005.

[12] Henk C. Tijms. *A First Course in Stochastic Models*. Wiley, second edition, 2003.